A Developers Guide to the Reservation Galaxy

Anto Loyola

(anto-nyu-ost-final-project.appspot.com)

Source Code :
github.com/antoloyola90/NYU_OST_Final_Project

Data Models

1. *Resource*

uniqueID = ndb.StringProperty(indexed=True)

The unique generated ID used to refer to this resource

user = ndb.UserProperty()

The user that owns this resource

name = ndb.StringProperty(indexed=False)

The name of the resource

startHour = ndb.IntegerProperty(indexed=False)

The starting hour of the resource availability

startMinute = ndb.IntegerProperty(indexed=False)

The starting minute of the resource availability

endHour = ndb.IntegerProperty(indexed=False)

The ending hour of the resource availability

endMinute = ndb.IntegerProperty(indexed=False)

The ending minute of the resource availability

tags = ndb.StringProperty(repeated=True)

The tags that are associated with this resource

2. *Reservation*

uniqueID = ndb.StringProperty(indexed=True)

The unique generated ID used to refer to this reservation

user = ndb.UserProperty()

The user that owns this resource

resourceUniqueID = ndb.StringProperty(indexed=True)

The unique generated ID used to refer to this resource that this reservation is for

startHour = ndb.IntegerProperty(indexed=False)

The starting hour of the reservation booking

startMinute = ndb.IntegerProperty(indexed=False)

The starting minute of the reservation booking

endHour = ndb.IntegerProperty(indexed=False)

The ending hour of the reservation booking

endMinute = ndb.IntegerProperty(indexed=False)

The ending minute of the reservation booking

dateDay = ndb.IntegerProperty()

The calendar day of the reservation booking

dateMonth = ndb.IntegerProperty()

The calendar month of the reservation booking *dateYear = ndb.IntegerProperty()*

The calendar year of the reservation booking

name = ndb.StringProperty(indexed=False)

The name of the resource that this reservation is for; used for display purposes

duration = ndb.IntegerProperty(indexed=False)

The duration in minutes of the reservation booking

Design of the User Interface

1. *Main Page – Class MainPage*

If there is no current user, redirects to login page.
Main Page has 5 parts:

Users' Reservations:

Displays the upcoming reservations, which the current user has made. It displays the name of the resource that is reserved, the booking date, the booking time with the duration and also gives an option to delete the reservation (links to Delete Reservation page). The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

***All Resources Available on the System**:*

Displays all the resources available on the system. . It displays the name of the resource, the availability hours and the tags associated with the resource. The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Resources owned by the User:

Displays all the resources available on the system. . It displays the name of the resource, the availability hours and the tags associated with the resource. The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Add a Resource Button:

Links to the Add Resource page

2. *Add/Edit Resource Page – Class AddResource*

If there is no current user, redirects to login page.
Add Resource Page takes 5 inputs:

Resource Name:

The name of the resource being added/edited

Start Time:

The starting availability time of the resource being added/edited between (00:00 and 23:59 in 5 minute increments)

End Time:

The ending availability time of the resource being added/edited (00:00 and 23:59 in 5 minute increments)

Tags:

User can enter a comma separated list of tags

Submit:

When the user clicks on this, the resource validation is done and if successful the resource is created/updated.

If there is any error then the page is reloaded with the values already entered by the user and the Error message is displayed on top.

3. *Resource Page – Class ViewResource*

If there is no current user, redirects to login page.
View Resource Page has 4 parts:

Resource:

It displays the name of the resource, the availability hours and the tags associated with the resource. The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Resource Reservations:

Displays the upcoming reservations, which the any user has made on the resource. It displays the user that made the reservation, the booking date, the booking time with the duration and also gives an option to delete the reservation (links to Delete Reservation page). The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Add Reservation Button:

Links to the Add Reservation page

Edit Resource Button:

Links to the Edit Resource page

4. *Add Reservation Page – Class AddReservation*

If there is no current user, redirects to login page.
Add Reservation Page takes 5 inputs:

Resource Name:

The name of the resource being added/edited

Date of Reservation:

The date that the reservation is requested

Start Time:

The starting availability time of the resource being added/edited between (00:00 and 23:59 in 5 minute increments)

Duration:

The duration in minutes of the reservation

Submit:

When the user clicks on this, the reservation validation is done and if successful the reservation is created.

If there is any error then the page is reloaded with the values already entered by the user and the Error message is displayed on top.

5. *View Tags Page – Class ViewTags*

If there is no current user, redirects to login page.
View Tags Page has 4 parts:

All Resources under tag:

> It displays all the resources that have the tag clicked on, the availability hours and the tags associated with the resource. The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

6. *RSS Generator Page – Class RSSGenerator*

If there is no current user, redirects to login page.
RSS Generator Page creates the xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
        <rss version="2.0">
              <title>library</title>
              <link>
                  /resource?resourceUniqueID=fde0ab99-af7b-
4636-ad0c-623ac176530e
            </link>
              <Resource>
                    <ResourceName>library</ResourceName>
                    <ResourceOwner>
                        al4251@nyu.edu
                  </ResourceOwner>
                  <NumberOfTimesReserved>
                        1
                  </NumberOfTimesReserved>
                   <ResourceTags>
                        <ResourceTag>apple</ResourceTag>
                        <ResourceTag>bob</ResourceTag>
                        <ResourceTag>map</ResourceTag>
                        <ResourceTag>uncle</ResourceTag>
                  </ResourceTags>
            <Reservations>
                  <Reservation>
                        <ReservedBys>
                              al4251@nyu.edu
                        </ReservedBy>
                       <StartTime>
                              <Format>
                                    hh:mm:ss--DD/MM/YY
                              </Format>
                              <Value>
                                    6:55:00--17/12/2016
                              </Value>
                       </StartTime>
                       <EndTime>
                              <Format>
                                    hh:mm:ss--DD/MM/YY
                              </Format>
                              <Values>
                                    7:0:00--17/12/2016
                              <Value>
                       </EndTime>
                    <Reservation>
```

```
                          <Reservation>
                                 <ReservedBys>
                                           al4251@nyu.edu
                                 </ReservedBy>
                                <StartTime>
                                           <Format>
                                                     hh:mm:ss--DD/MM/YY
                                           </Format>
                                           <Value>
                                                     7:5:00--17/12/2015
                                           </Value>
                                </StartTime>
                                <EndTime>
                                           <Format>
                                                     hh:mm:ss--DD/MM/YY
                                           </Format>
                                           <Values>
                                                     7:10:00--17/12/2015
                                           <Value>
                                </EndTime>
                         <Reservation>

              </Reservations>
    </Resource>
</rss>
```

7. User Page – Class MainPage

If there is no current user, redirects to login page.
User Page has 2 parts:

Users' Reservations:

Displays the upcoming reservations, which the current user has made. It displays the name of the resource that is reserved, the booking date, the booking time with the duration and also gives an option to delete the reservation (links to Delete Reservation page). The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Resources owned by the User:

Displays all the resources available on the system. . It displays the name of the resource, the availability hours and the tags associated with the resource. The name of the resource has an RSS link that generates its xml with the reservations on the resource (links to RSS Generator page).

Implementation Assumptions

- Resources can have the same name
- Resources can only have reservations on the same day
- In RSS, even the previous reservations that have passed are printed to help with book-keeping
- If an owner edits a resource then the reservations already on the system will not change and changes are made only for future reservations
- The Resource once created, cannot be deleted but can be changed to another resource. This functionality would be very easy to implement but I did not feel that I would learn anything new by implementing this. Concentrated on the extra credits instead ☺

Extra Credit

- An Email will be sent out when the Reservation is made

- An Email will be sent out when the Reservation start time is 5 minutes away. This was implemented using a Cron Job.

- Resources can be searched by Name. Partial hits will also be shown. If no search value is given by the user then all resources are displayed

- Resources can be searched by Availability. Takes the Date, Start Time and the duration that the user is searching for. If duration is entered incorrectly by the user then duration = 1 hour.

- The number of times a resource has been reserved in the past is also shown wherever the resource is displayed

- Reservations cannot be made if there is already a reservation by the user for any resource during that time