

Open Source Tools
GA.2246-001: Spring 2017
ReserveBook

<https://reservebook-ost.appspot.com/>

Developers Guide
Jatin Malviya
jm6474@nyu.edu

Code: <https://github.com/JatinMalviya/ReserveBook-Reservation-Application>

Features Implemented:

- ✓ Handle multiple users and each user should be able to create resources or reserve them.
- ✓ On the landing page, viewers will see three sections: (1) reservations made for resources by that user (2) all resources in the system, shown in reverse time order based on last made reservation; (3) resources that the user owns, each linked to its URL; and (4) a link to create a new resource
- ✓ Once the end time for a reservation has passed, it should not be shown anymore in any view.
- ✓ When a resource is created, it should allow the following data about a resource to be entered: the name, availability hours, and a set of tags that describe the type of resource.
- ✓ Each resource should have a URL that can be used to (1) show the current and upcoming reservations for a resource; (2) add a new reservation for the resource; and (3) edit information about the resource, only if the user viewing the resource is the owner. When editing, the previously entered information about the resources is shown in the form as default values. Anywhere a resource name is shown, it should link to the corresponding URL.
- ✓ Reservations can be made by specifying a start time a duration for the resource. You do not need to support reservations that span a day boundary. If a user tries to make a reservation for a resource at a time where it is not available, or is not within the available hours of the resource, the UI will not accept the reservation and tell the user why.
- ✓ Each reservation should display the user, resource name, reservation time and duration. The resource name should be linked to the resource page, and the user should be linked to a page that shows all the reservations and resources for a user.
- ✓ Existing reservations can be deleted by the user who made the reservation
- ✓ Each resource should support the notion of having tags. When viewing resources, the tags for the resource will be shown, and each tag links to a URL that shows all the resources with the given tag.
- ✓ Each resource will have an RSS link that dumps all reservations for that resource in XML format (see wiki page for example).
- ✓ Option to turn off the RSS link in the project.

Extra Credits:

- ✓ When showing a resource, show how many times it has been reserved in the past.
- ✓ Support the ability to search for resources by name.
- ✓ Support the ability to search for resources by availability (user enters a time and duration, all resources available during those times are displayed).
- ✓ When reservations are made, an email confirmation would be sent to a user.
- ✓ When a reservation starts, an email confirmation would be sent to a user.
- ✓ Support a requirement where a user can only have one reservation at a time. The system would not allow a reservation to be made if it overlaps with another one made by the same user.

Notes / Design Decisions:

- At times when the expected behavior was not clearly mentioned in the problem statement, I have made assumptions which are mentioned below.
- All times are in EDT.
- Resources on landing page are sorted by last reservation time i.e. the time when a booking was made on that resource.
- Reservations are sorted by first start time and then end time.
- A reservation will be shown till its end time has not been passed.
- Wherever a resource name is shown, it links to that resource's page.
- Wherever a user's email is shown, it links to that user's page.
- Tags are read as space separated.
- Whenever a resource is created, its last reservation time field is set to its creation time so that the new resource comes at top in the resources list on the landing page.
- Resource Names can be duplicate, but will have different resource ids in the system.
- Whenever a resource is edited, the past & upcoming resources are not affected. They remain as it is because it will result in bad user experience if there reservations are deleted without any intimation.
- A user cannot make a reservation if he/she already has a reservation for that interval.
- A user cannot book a resource for a time interval if the resource is already booked by someone else in that time interval (fully or partly).
- A user can see any other user's page but can delete only his reservations.
- A Reservation owner can only delete the reservation.
- A Resource owner can only edit the resource.
- The total reservations in resources field show the total number of past and upcoming reservations for that resource.
- RSS_FLAG can be used to turn off/on the RSS feed feature.
- RSS Feed is off in the experiment branch.
- RSS feed for a resource shows all its properties and all the reservations (past and present).
- Both search by name and availability (start time & duration) are implemented.

- Search by name is case insensitive & if a resource contains the search term anywhere in its name, it will be returned.
- There is a limit of 10 emails each day in the free version of Google App Engine. After this limit is reached, the system stops sending mails for that day. So there's a possibility that emails might not be sent.
- Emails are sent with the sender's address as: <ReserveBook@reservebook-ost.appspotmail.com>
- There is a cron job in the system which runs every minute to send mails informing start of the reservation.
- Bootstrap is used for most of the styling.

Data Models:

Model: Resource

Used to store information about all the resources in the system.

<i>Field</i>	Type	Description
<i>resourceId</i>	ndb.StringProperty	Unique Id for each resource.
<i>resourceName</i>	ndb.StringProperty	Resource's Name
<i>startTime</i>	ndb.TimeProperty	Start Time of Resource's Availability
<i>endTime</i>	ndb.TimeProperty	End Time of Resource's Availability
<i>tags</i>	ndb.StringProperty	List of Tags that describe type of Resource
<i>owner</i>	ndb.StringProperty	Creator of the Resource
<i>lastReservationTime</i>	ndb.DateTimeProperty	Last time when a Reservation was made on the Resource. (Creation time if no reservation)
<i>count</i>	ndb.IntegerProperty	Number of Past and Upcoming Reservations for the Resource.

Model: Reservations

Used to store information about all the reservations in the system.

<i>Field</i>	Type	Description
<i>reservationId</i>	ndb.StringProperty	Unique Id for each reservation.
<i>resourceId</i>	ndb.StringProperty	Id of the Resource for which the Reservation is made.
<i>resourceName</i>	ndb.StringProperty	Name of the Resource for which the Reservation is made.
<i>startTime</i>	ndb.DateTimeProperty	Start Time of the Reservation
<i>endTime</i>	ndb.DateTimeProperty	End Time of the Reservation
<i>duration</i>	ndb.TimeProperty	Duration of the Reservation
<i>reservationTime</i>	ndb.DateTimeProperty	Time when the Reservation was made.
<i>user</i>	ndb.DateTimeProperty	User who made the reservation.

System Modules:**1. Landing Page**

Url: <https://reservebook-ost.appspot.com/>

Template: Index.html

Shows current user's upcoming reservations (sorted by lastReservationTime field), all the resources (sorted by availability start time and end time), user owned resources (sorted by availability start time and end time) and options for creating and search resources. Each resource name links to that resource's pages and user email links to that user's page. Also, user has option to see RSS feed of the resource and delete his reservations.

2. Resource Page

Url: <https://reservebook-ost.appspot.com/Resource?resourceId=<resourceId>>

Template: Resource.html

Shows all the fields (information) about the resource, reservations for that resource (both ongoing and upcoming) and option to create reservation for that resource. Also, user can edit the resource from here only if he is the owner of that resource.

3. User Page

Url: <https://reservebook-ost.appspot.com/User?<userEmail>>

Template: User.html

Shows all the reservations (ongoing and upcoming) and resources owned by the user. If current user is owner of this profile, he can delete the reservation from here.

4. Tag Page

Url: <https://reservebook-ost.appspot.com/Tag?tag=<tag>>

Template: Tag.html

Shows all the resources with the given tag.

5. Add Resource Page

Url: <https://reservebook-ost.appspot.com/AddResource>

Template: AddResource.html

Creates a new resource with the entered information. Resource name, availability and tags are asked from the user. Proper validation checks are done on the input – format check, validity of times entered and checking if end time is after start time. Then a unique id is generated and a resource is created after parsing all the fields. The count field of the resource is set to 0 and lastReservationTime is set to creation time.

6. Edit Resource Page

Url: <https://reservebook-ost.appspot.com/EditResource?resourceId=<resourceId>>

Template: EditResource.html

Shown only if the current user is the owner of the resource. All the fields of resource of the resource are displayed, giving option to the user to change any of them. All the validations of “Add Resource Page” are done here too. lastReservationTime is changed to the editing time. Note: the existing reservations for the resource are not changed.

7. Add Reservation Page

Url: <https://reservebook-ost.appspot.com/AddReservation?resourceId=<resourceId>>

Template: AddReservation.html

Provides functionality to add reservation for the resource. The start time and duration for the reservation is asked and then all the validations are done before adding the reservations: checking format of the data entered, checking dates, checking validity of times, booking date not in past, checking if resource is available at that time, checking if resource is free at that time and checking if user is free at that time. When successful, the resource’s count is increased and a mail is sent to the user. Also the lastReservationTime for resource is changed to current time.

8. Delete Reservation Page

Url: <https://reservebook-ost.appspot.com/DeleteReservation?reservationId=<reservationId>>
Template: DeleteReservation.html

Shows all the information about the reservation and asks user to confirm the deletion. If confirmed the reservation removed and the reservation's resource booking count is decreased.

9. Generate RSS Page

Url: <https://reservebook-ost.appspot.com/GenerateRss?resourceId=<resourceId>>
Template: Rss.html

Dump all the information and all the reservations (past, current & upcoming) in generic XML format. Depends on the value of RSS_FLAG in the code.

10. Search By Name Page

Url: <https://reservebook-ost.appspot.com/SearchName>
Template: SearchName.html

Asks use to enter the search term and then all the resources which contain that term in their names are displayed to the user.

11. Search By Availability Page

Url: <https://reservebook-ost.appspot.com/SearchAvailability>
Template: SearchAvailability.html

Asks use to enter the start time & duration for a time interval and then all the resources which are open (have available hours) during that interval displayed to user. Note: this doesn't mean that the resource is actually free in that interval, it may happen that the resource id booked during that interval by someone else. It just shows resources that are open to booking during that interval.

12. Send Reminder Mail

Url: <https://reservebook-ost.appspot.com/SendReminderMail>
Template: None

This is not accessible by user. This is used by the cron job every minute to check if there is a reservation starting now, if there is, a reminder mail is sent to the owner of the reservation.

References:

- <https://cloud.google.com/appengine/docs/>
- <https://www.w3schools.com/bootstrap>
- <https://v4-alpha.getbootstrap.com>
- *Python Docs*

