**Restaurants Reservation back-end service:**

**Abstract:**

There will be already guest, restaurant and table data will be available in DB.

So we do reservation of tables of restaurant with number of guests and reservation time by our API.

Here It have suppose the table will booked for 1 hour from time of reservation to keep application simple due to lack of time for development.

e.g. suppose reservation time is 22 Jan, 2019 10:00 am

So the table will be not allow booking between above time range.

It have considered most of the scenarios and the validation you have mentioned.

**Application is deployed on heroku:** https://restaurant-services.herokuapp.com/

**Note:**

* Please have a look at screen-shots for more information.
* Refer to **README.md** for local application setup steps.
* I have added project doment file into source code **Restaurants Reservation backend service.docx** and also into the code task completion email.

**GitHub public repository link:**

https://github.com/akkisharma/umai\_reservation\_services

**Heroku(production) version of code:** commited to **heroku\_changes** branch.

**API:**

**1: create a reservation**

URL: https://restaurant-services.herokuapp.com/api/v1/reservations

method: POST

params:

{

"guest\_id" : 1,

"restaurant\_id" : 1,

"table\_id": 1,

"guest\_count" : 4,

"reservation\_time" : "22 jan 2019 12:00 pm"

}

**2: updates the reservation**

URL: https://restaurant-services.herokuapp.com/api/v1/reservations/<reservation\_id>

method: PUT

params:

{

"reservation\_time" : "23 jan 2019 12:00 pm",

"guest\_count" : 5

}

**3: list of reservations for a restaurant**

URL: https://restaurant-services.herokuapp.com/api/v1/restaurants/<restaurant\_id>

Method: GET

**Run RSpec:**

**$ bundle exec rspec --format documentation**

Api::V1::ReservationsController

POST #create

with valid parameters

creates a new reservation

with invalid parameters

does not create a new reservation

Api::V1::RestaurantsController

GET #show

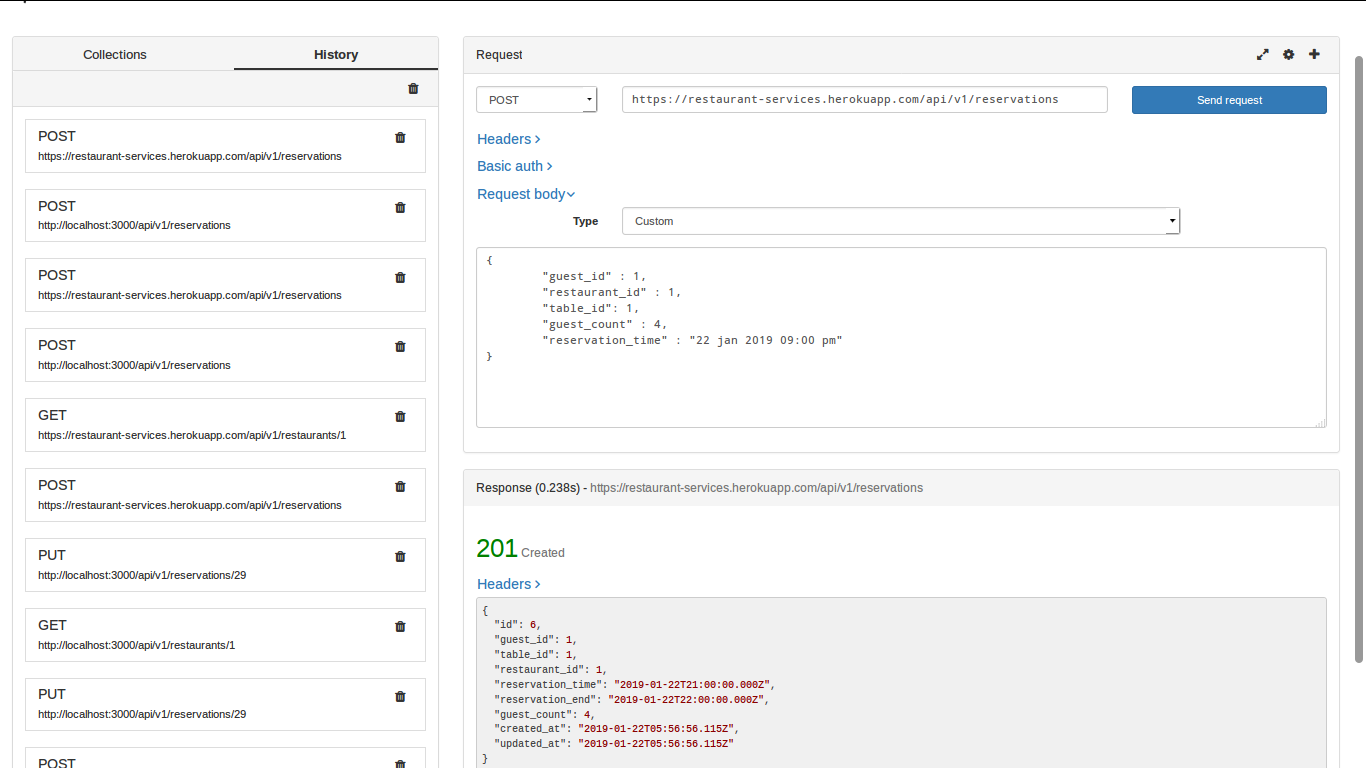
with valid parameters

returns http success

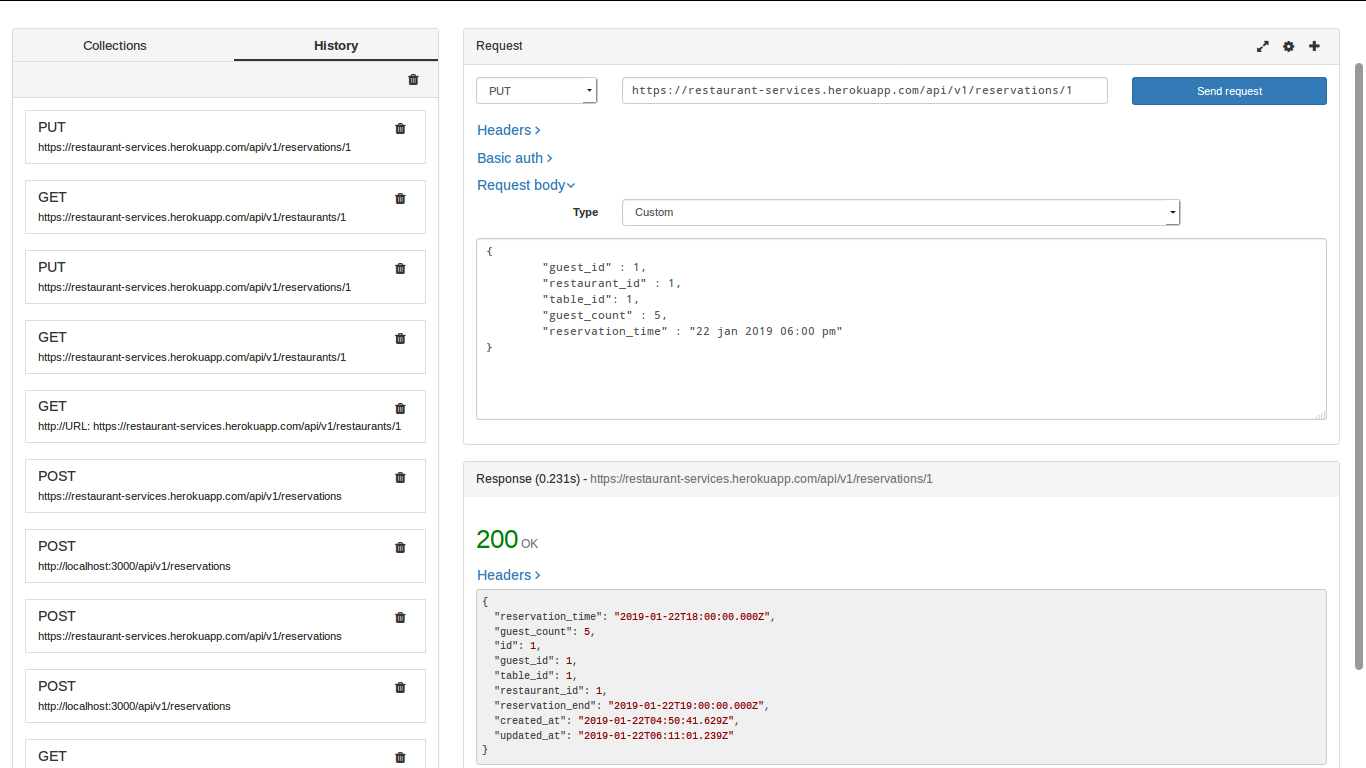
Finished in 1.05 seconds (files took 6.19 seconds to load)

3 examples, 0 failures

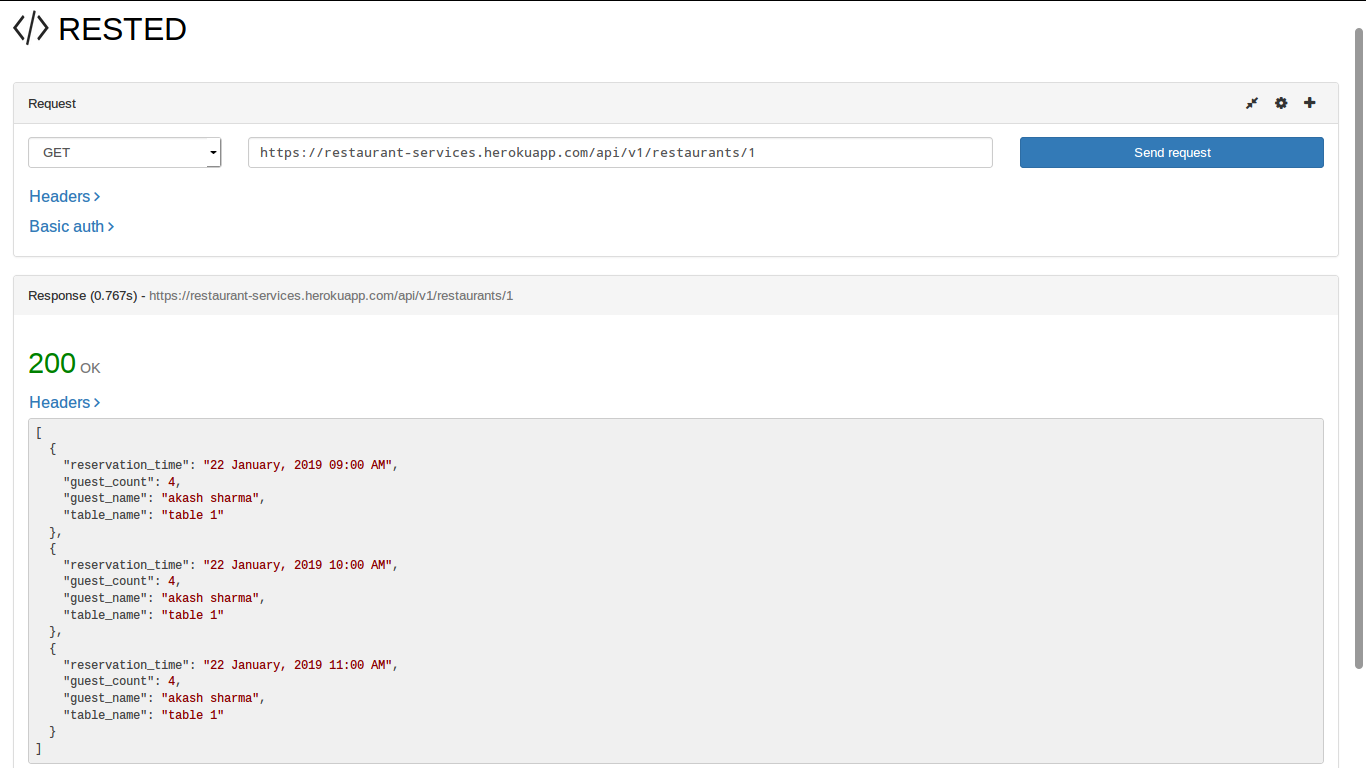
**API requests calling screen-shots:**



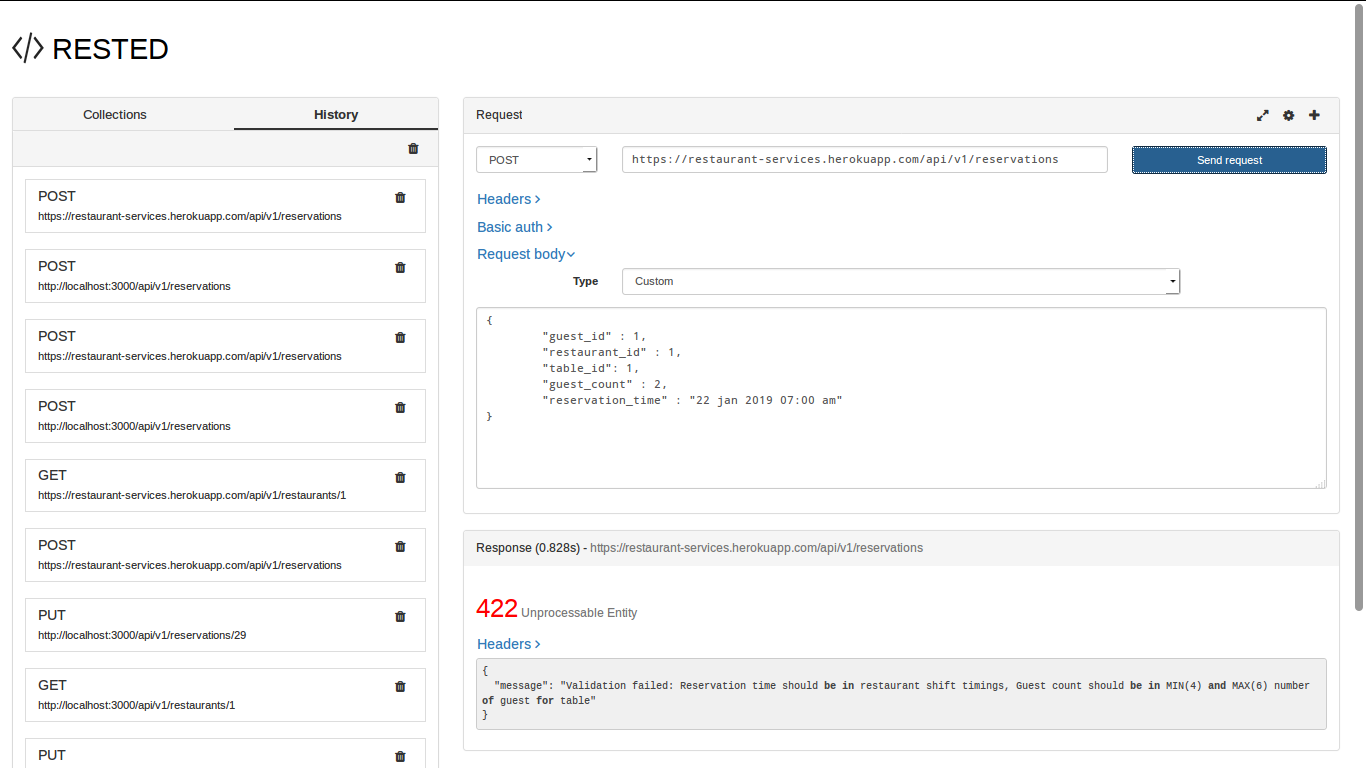
2.



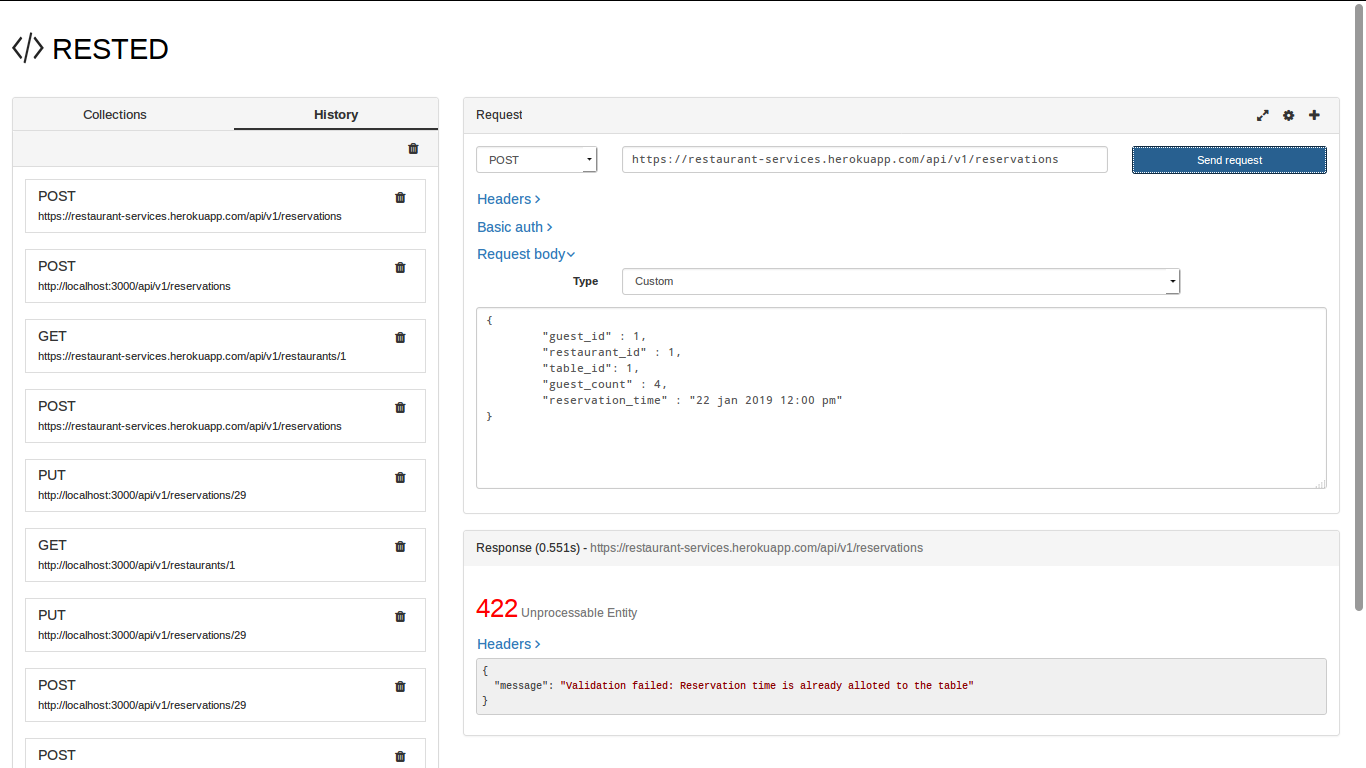
3.



4.

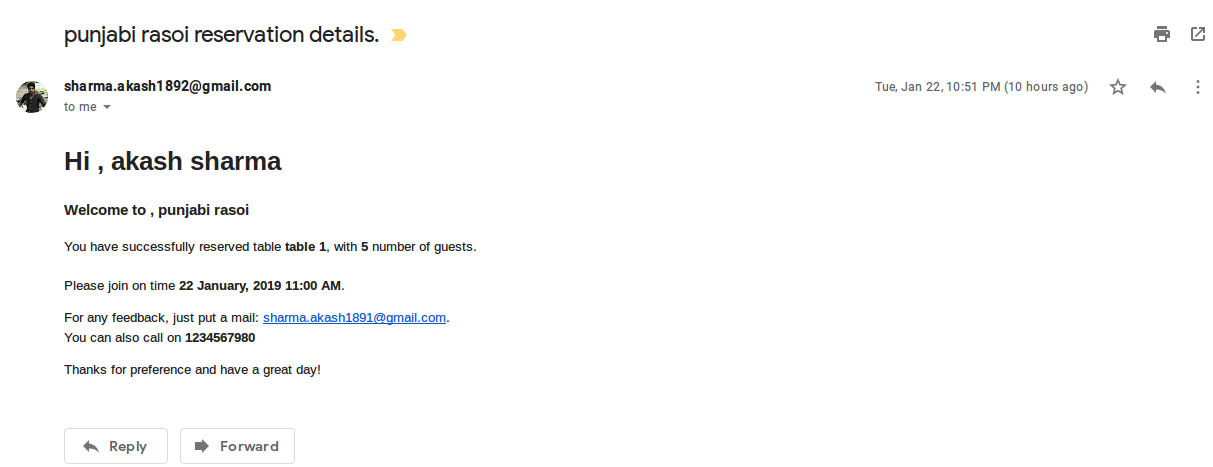


5.

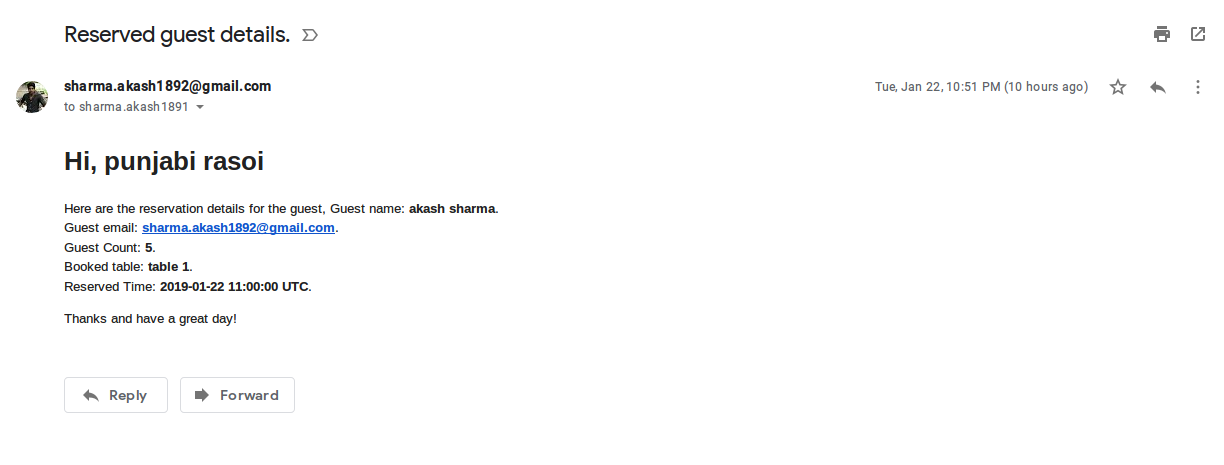


**Mail Template:**

1. Mail to guest:

****

1. Mail to restaurant:

****