Models:

class Property(models.Model):

name

location

description

number\_of\_guests

number\_of\_beds

area: How big is the house

avaliability: Dates avaliable to be reserved

avaliable\_to\_reserve: Bool, is it listed to be reserved?

image

owner

class ReservationRequest(models.Model):

user

start\_date

end\_date

property\_id: Id of the property being requested to reserve

status: Approved, Cancelled, terminated, etc

Endpoints:

* userpropertylist/

Method: GET

Description: View all user property

* create\_property/

Method: GET + POST

Description: Create property for user

Payload: ["name", "location", "description", "number\_of\_guests", "number\_of\_beds", "area", "avaliability", "image", "owner"]

* get\_property/<int:id>/

Method: GET

Description: Get a property based on its id

Payload: ['id',"name", "location", "description", "number\_of\_guests", "number\_of\_beds", "area", "avaliability", "avaliable\_to\_reserve", "owner"]

* update\_property/<int:id>/

Method: GET + PUT

Description: Update a property based on its id

Payload: ["name", "location", "description", "number\_of\_guests", "number\_of\_beds", "area", "avaliability", "avaliable\_to\_reserve"]

* delete\_property/<int:id>/

Method: GET + DELETE

Payload: ["name", "location", "description", "number\_of\_guests", "number\_of\_beds", "area", "avaliability", "owner"]

Description: Delete a property based on its id

* all\_property/

Method: GET

Description: Get all property created by all users

* user\_reservations\_host/

Method: GET

Description: Get all property listed by user to be reserved by others

* allreservations/

Method: GET

Description: Get all property listed by ALL users to be reserved by others

* create\_reservation/

Method: :GET + POST

Description: Create a reservation request for a listed property

Payload: ['start\_date', 'end\_date', 'property\_id', 'user']

* user\_reservations/

Method: GET

Description: Get all list of user's reservation requests

* delete\_user\_reservations/<int:propertyid>/

Method: GET + DELETE

Description: Delete user's reservation request to a certain property using its property id

Payload: ['start\_date', 'end\_date', 'property\_id', 'status', 'user']

* view\_reservation\_requests/

Method: GET

Description: List of reservation requests to any of user's property that is avaliable to reserve

* update\_reservation\_requests/<int:resv\_id>/

Method: GET + PUT

Description: Change the status of reservation requests to any of user's property that is avaliable to reserve

Payload: ['status']

**Models**:

class Comment(models.model):

author - identifier of user writing the comment

property\_id – corresponding ID of the property that the user is commenting on

name – name of user writing the comment

title

date

content

class CommentForUser(models.model):

author - identifier of user writing the comment

recipient – name of the user (guest) that the author (property owner) is writing about

name - name of user writing the comment

title

date

content

**Endpoints**:

* create\_comment/  
  Method: GET + POST  
  Payload: ['author', 'property\_id', 'name', 'title', 'date', 'content',]  
  Description: Create a new comment for a user (guest) to write about a property/owner
* get\_comment/<int:id>/  
  Method: GET  
  Payload: ['id', 'author', 'property\_id', 'name', 'title', 'date', 'content',]  
  Description: Get a comment based on its id
* update\_comment/<int:id>/  
  Method: GET + PUT  
  Payload: ['title','date', 'content',]  
  Description: Update a comment based on its ID
* delete\_comment/<int:id>/  
  Method: GET + DELETE  
  Payload: ['author','title','date', 'content',]  
  Description: Delete a comment based on its ID
* all\_comment/  
  Method: GET  
  Description: Lists all the comments written about properties and their owners
* list\_comment/   
  Method: GET  
  Description: List all the comments written by a particular user (guest)
* create\_commentforuser/  
  Method: GET + POST  
  Payload: ['author', 'recipient', 'name', 'title', 'date', 'content',]  
  Description: Create a new comment for a user(property owner) to write about another user (guests)
* get\_commentforuser/<int:id>/  
  Method: GET  
  Payload: ['id', 'author', 'recipient', 'name', 'title', 'date', 'content',]  
  Description: Get a comment based on its id
* update\_commentforuser/<int:id>/  
  Method: GET + PUT  
  Payload: ['title','date', 'content',]  
  Description: Update a comment based on its ID
* delete\_commentforuser/<int:id>/  
  Method: GET + DELETE  
  Payload: ['author','title','date', 'content',]  
  Description: Delete a comment based on its ID
* all\_commentforuser/  
  Method: GET  
  Description: Lists all the comments written about guests/people who make reservations
* list\_commentforuser/   
  Method: GET  
  Description: List all the comments about other users(guests) written by a particular user (property owner)