Project APIs for Rooms/Events

Andres Loor - August 2022

These APIs were created Python/Django considering Model View - Template (MVT) and the default Database was used for this Project (SQLite 3).

Database URL: http://127.0.0.1:8000/admin/

username: andresloor

password: 12345678

It was required to build 4 classes:

Class Room

ID

Name

Description

Capacity

Class Event

ID

Name

Description

Type

Date

Room

Available Places

Class Customer

ID

Name

Description

Class Booking

ID

Description

Customer

Event

Status

CRUD Operations

- GET ROOM LIST
  - http://127.0.0.1:8000/rooms/

- GET ROOM DETAIL
  - http://127.0.0.1:8000/rooms/1

- POST NEW ROOM
  - http://127.0.0.1:8000/rooms/
  - Body(Name, Description, Capacity)

- PUT UPDATE ROOM
  - http://127.0.0.1:8000/rooms/3
  - Body(Name, Description, Capacity)

- DELETE ROOM
  - http://127.0.0.1:8000/rooms/7

- GET EVENT LIST
  - http://127.0.0.1:8000/events/

- GET EVENT DETAIL
  - http://127.0.0.1:8000/events/1

- POST NEW EVENT
  - http://127.0.0.1:8000/events/
  - Body(Name, Description, Type, Date, Room, Available Places)

- PUT UPDATE EVENT
  - http://127.0.0.1:8000/events/1
  - Body(Name, Description, Type, Date, Room, Available Places)
  -
- DELETE EVENT
  - http://127.0.0.1:8000/events/2

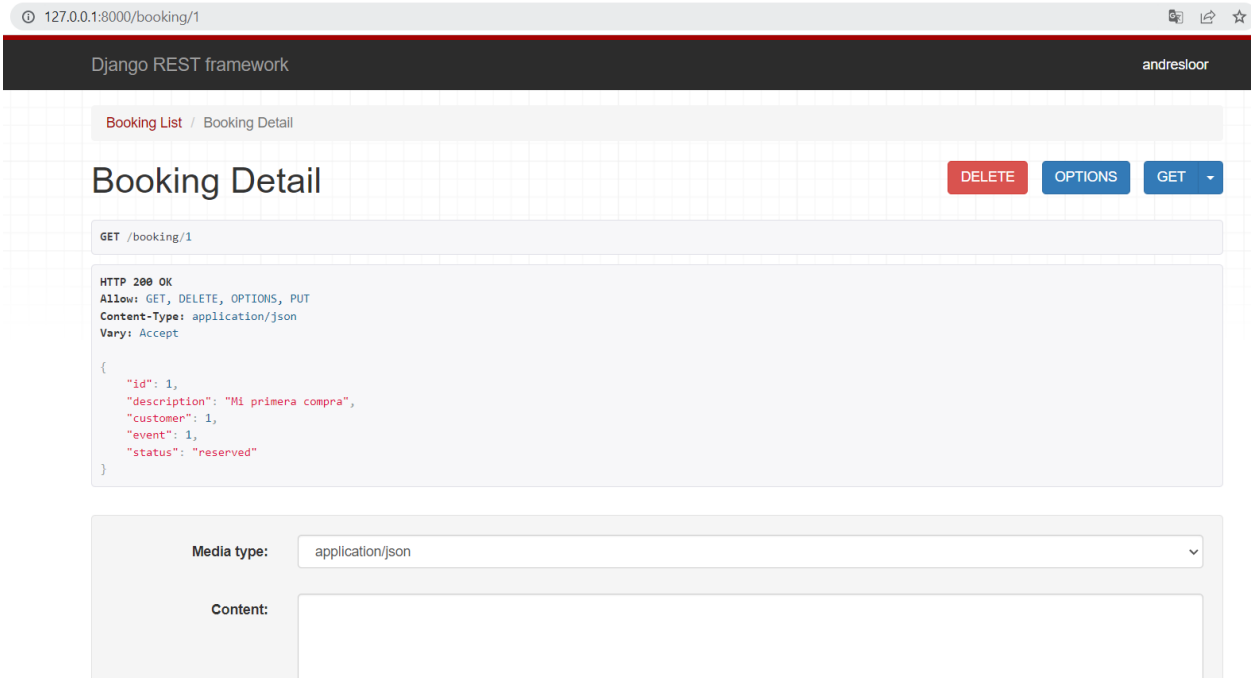
- GET BOOKING LIST
  - http://127.0.0.1:8000/booking/

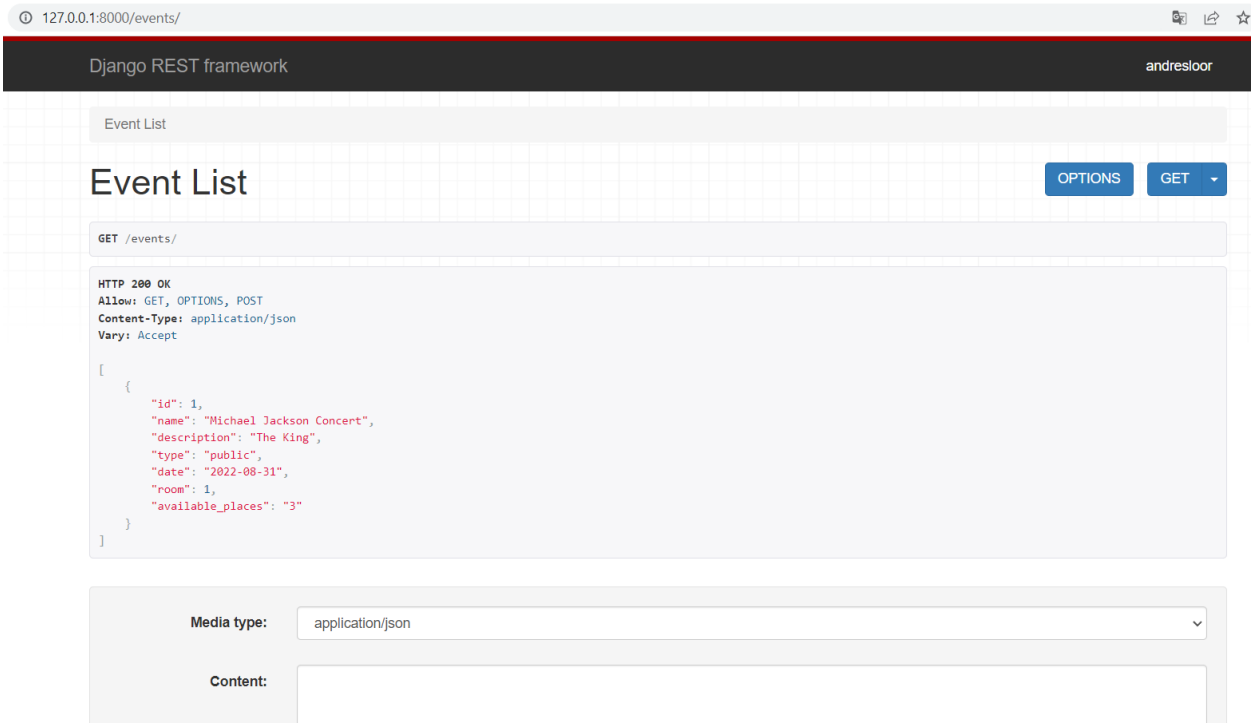
- GET BOOKING DETAIL
  - http://127.0.0.1:8000/booking/1

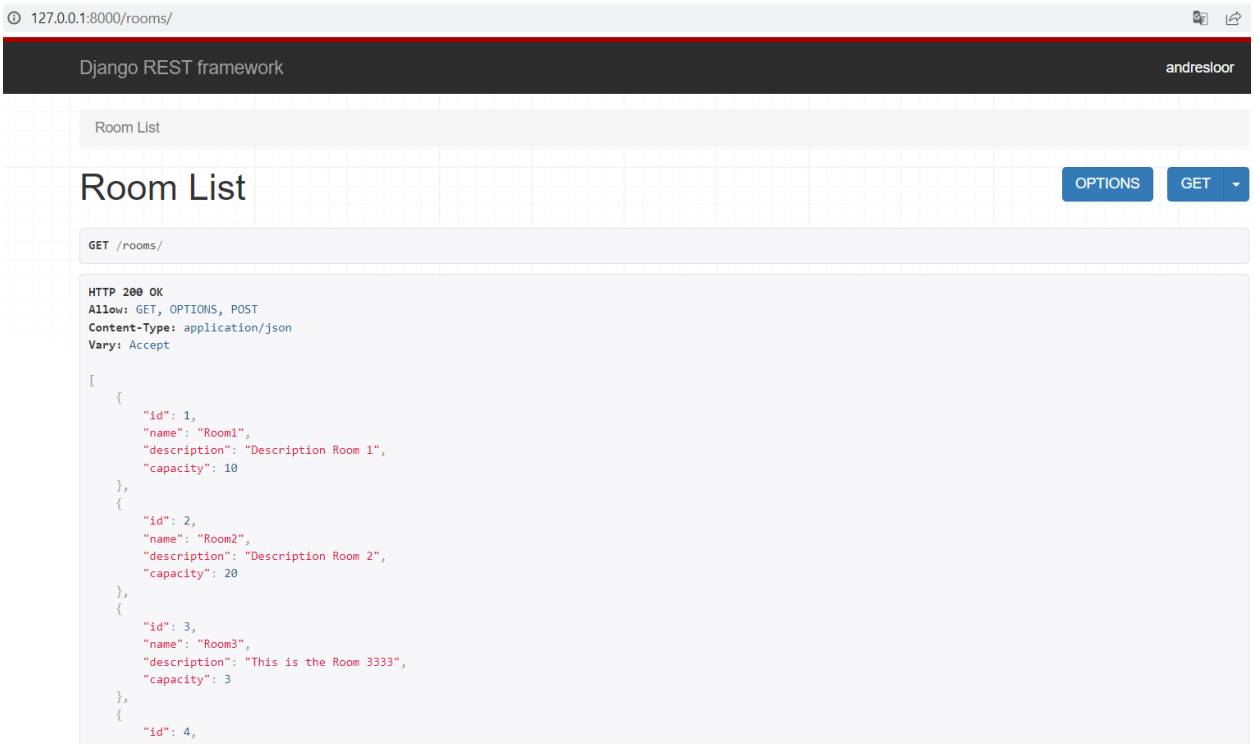
- POST NEW BOOKING
  - http://127.0.0.1:8000/booking/
  - Body(Description, Customer, Date, Event, Status)

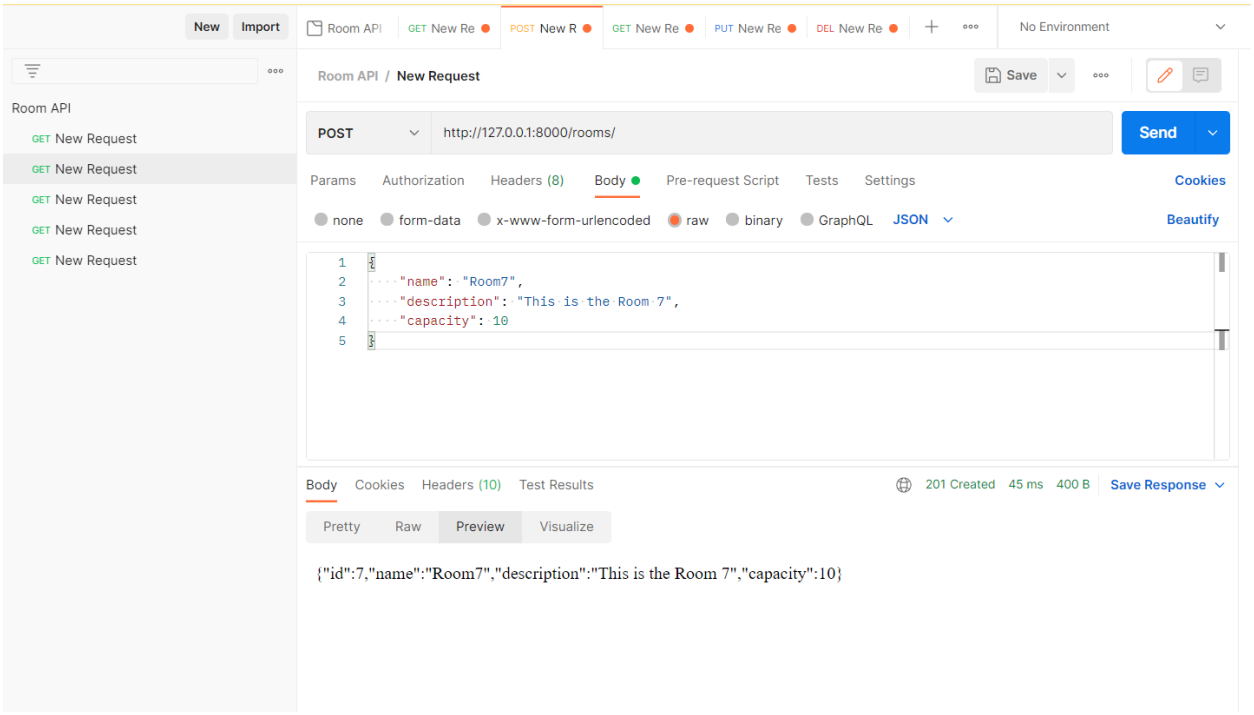
- PUT UPDATE BOOKING
  - http://127.0.0.1:8000/booking/1
  - Body(Description, Customer, Date, Event, Status)
  -
- DELETE BOOKING
  - http://127.0.0.1:8000/booking/2

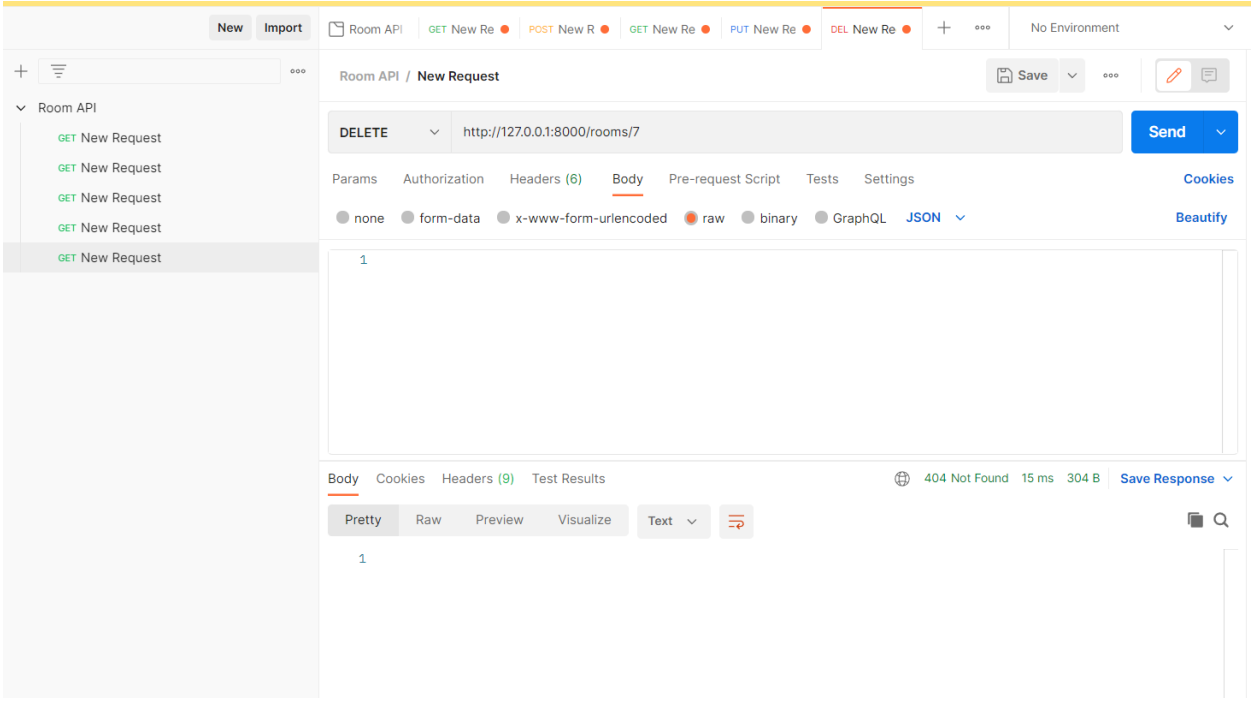
EXAMPLES of how API works




127.0.0.1:8000/events/
Django REST framework
andresloor
Event List
Event List
OPTIONS
GET
GET /events/
HTTP 200 OK
Allow: GET, OPTIONS, POST
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"name": "Michael Jackson Concert",
"description": "The King",
"type": "public",
"date": "2022-08-31",
"room": 1,
"available_places": "3"
}
]
Media type:
application/json
Content:

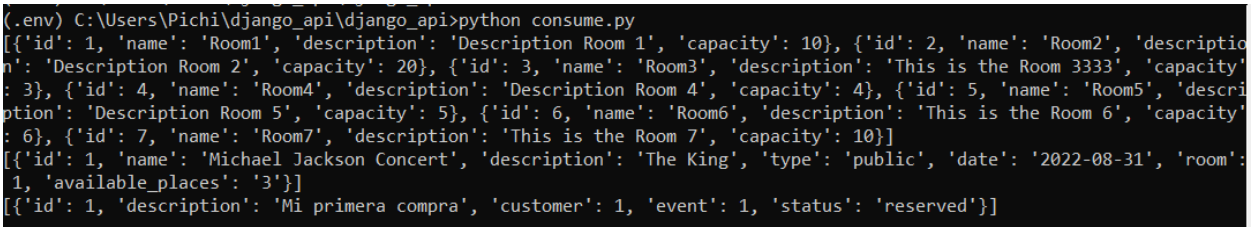

127.0.0.1:8000/rooms/
Django REST framework
andresloor
Room List
Room List
OPTIONS
GET
GET /rooms/
HTTP 200 OK
Allow: GET, OPTIONS, POST
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"name": "Room1",
"description": "Description Room 1",
"capacity": 10
},
{
"id": 2,
"name": "Room2",
"description": "Description Room 2",
"capacity": 20
},
{
"id": 3,
"name": "Room3",
"description": "This is the Room 3333",
"capacity": 3
},
{
"id": 4,


New
Import
Room API
GET New Re
POST New R
GET New Re
PUT New Re
DEL New Re
No Environment
Room API
GET New Request
GET New Request
GET New Request
GET New Request
GET New Request
Room API / New Request
Save
POST
http://127.0.0.1:8000/rooms/
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "Room7",
"description": "This is the Room 7",
"capacity": 10
}
Body
Cookies
Headers (10)
Test Results
201 Created
45 ms
400 B
Save Response
Pretty
Raw
Preview
Visualize
{"id":7,"name":"Room7","description":"This is the Room 7","capacity":10}


New
Import
Room API
GET New Re
POST New R
GET New Re
PUT New Re
DEL New Re
No Environment
Room API
GET New Request
GET New Request
GET New Request
GET New Request
GET New Request
Room API / New Request
Save
DELETE
http://127.0.0.1:8000/rooms/7
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Body
Cookies
Headers (9)
Test Results
404 Not Found
15 ms
304 B
Save Response
Pretty
Raw
Preview
Visualize
Text

TESTS

The **custome.py** file runs test from the GET LIST classes.



The command is> python consume.py