

## 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

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/booking/1`. The page header indicates it's the Django REST framework, with the username `andresloor` in the top right. The main content area is titled "Booking Detail" and includes a breadcrumb "Booking List / Booking Detail". To the right of the title are three buttons: "DELETE" (red), "OPTIONS" (blue), and "GET" (blue with a dropdown arrow). Below the title, a box shows the HTTP method `GET /booking/1`. A larger box displays the response details: `HTTP 200 OK`, `Allow: GET, DELETE, OPTIONS, PUT`, `Content-Type: application/json`, and `Vary: Accept`. The response body is a JSON object: 

```
{  "id": 1,  "description": "Mi primera compra",  "customer": 1,  "event": 1,  "status": "reserved"}
```

. At the bottom, there's a section for "Media type" with a dropdown menu set to `application/json`, and a "Content:" label followed by an empty text area.

127.0.0.1:8000/events/

Django REST framework

andresloor

Event List

Event List

OPTIONSGET

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

OPTIONSGET

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,  
 "name": "Room4",  
 "description": "Description Room 4",  
 "capacity": 5  
 }  
]

NewImport

Room APIGET New RePOST New RGET New RePUT New ReDEL New Re+...No Environment

Room API / New Request

Save

Send

POST

http://127.0.0.1:8000/rooms/

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedorawbinaryGraphQLJSON

12345

```
1 {
2   ... "name": "Room7",
3   ... "description": "This is the Room 7",
4   ... "capacity": 10
5 }
```

BodyCookiesHeaders (10)Test Results

201 Created45 ms400 BSave Response

PrettyRawPreviewVisualize

```
{ "id": 7, "name": "Room7", "description": "This is the Room 7", "capacity": 10 }
```

NewImport

Room APIGET New RePOST New RGET New RePUT New ReDEL New Re+...No Environment

Room API / New Request

Save

Send

DELETE

http://127.0.0.1:8000/rooms/7

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedorawbinaryGraphQLJSON

1

BodyCookiesHeaders (9)Test Results

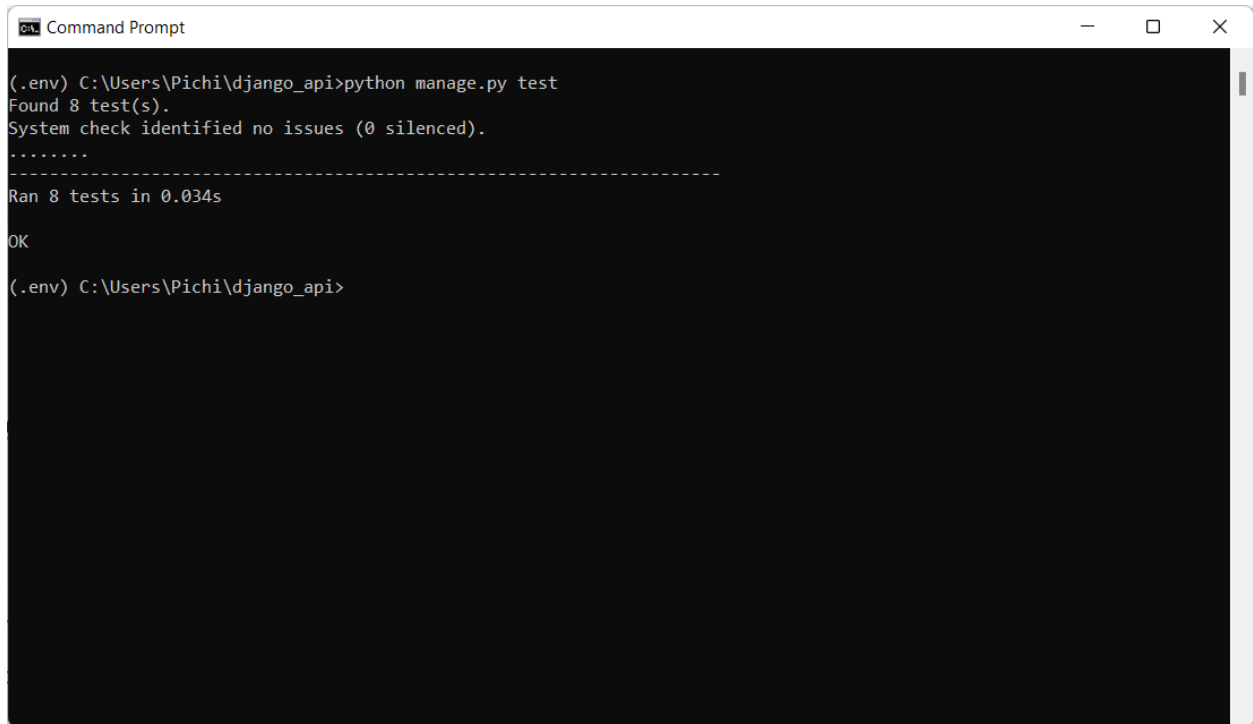
404 Not Found15 ms304 BSave Response

PrettyRawPreviewVisualizeText

1

## TESTS

The **test.py** file runs 8 basic test from all the APIs.



```
Command Prompt
(.env) C:\Users\Pichi\django_api>python manage.py test
Found 8 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 8 tests in 0.034s

OK
(.env) C:\Users\Pichi\django_api>
```

The command is `> python manage.py test`

The result must be OK if anything changes.

If you need to try the **APIs**, can use this command line:

`python consume.py`

```
Command Prompt
Volume Serial Number is 5E32-C3A6

Directory of C:\Users\Pichi\django_api

08/25/2022  04:30 AM  <DIR>          .
08/24/2022  01:41 AM  <DIR>          ..
08/23/2022  10:52 PM  <DIR>          .env
08/25/2022  04:30 AM              176,128 db.sqlite3
08/25/2022  04:27 AM  <DIR>          django_api
08/23/2022  11:02 PM              688 manage.py
08/24/2022  06:25 AM          392,654 README.docx
08/24/2022  06:25 AM          426,101 README.pdf
               4 File(s)          995,571 bytes
               4 Dir(s)  2,413,907,968 bytes free

(.env) C:\Users\Pichi\django_api>cd django_api

(.env) C:\Users\Pichi\django_api\django_api>python consume.py
[{'id': 1, 'name': 'Room1', 'description': 'Room 1', 'capacity': 10}, {'id': 2, 'name': 'Room2', 'description': 'This is the Room 2 Updated', 'capacity': 20}, {'id': 3, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 4, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 5, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 6, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 7, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 8, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}, {'id': 9, 'name': 'Room30', 'description': 'This is the New Room 30', 'capacity': 10}]
[{'id': 3, 'name': 'Michael Jackson Concert', 'description': 'King', 'type': 'public', 'date': '2022-09-01', 'room': 1, 'available_places': 2}]
[{'id': 2, 'description': 'Mi primera compra', 'customer': 1, 'event': 3, 'status': 'reserved'}]

(.env) C:\Users\Pichi\django_api\django_api>
```