

# **ЛУНАПАРКОВА ЛУДНИЦА - КОМАНДНО ТАБЛО**

**КАЗУС 2  
ДОКУМЕНТАЦИЯ**

## 1. ТЕРМИНОЛОГИЯ, КОЯТО СЕ СРЕЩА В ДОКУМЕНТА:

- URL – адреса, чрез който се достъпва дадения ресурс. Винаги започва с localhost:5000 или 127.0.0.1:5000. (localhost:5000/attractions/4)
- Метод – вид на заявката – дали е за създаване на нов запис (POST), редакция на съществуващ запис (PUT), изтриване на съществуващ запис (DELETE), или извличане на съществуващ запис (GET).
- Сървърен отговор – отговора, който сървърът връща, спрямо заявката, която е получила:

Код на отговор	Наименование	Описание
200	OK	Заявката е успешно обработена.
201	Created	Заявката е успешна; създаден е нов ресурс.
400	Bad Request	Заявката е непълна, неточна или грешна.
404	Not Found	Ресурсът не е намерен / не съществува.
500	Internal Server Error	Заявка, която не може да се обработи – грешка от сървърна страна.

## 2. ФОРМАТ НА ИЗПРАЩАНЕ И ПОЛУЧАВАНЕ НА ДАННИ

Интернет приложението работи с JSON формат. Всяка една заявка за извличане представя данните в JSON. При POST & PUT заявки се приемат данни в същия формат!

Пример за JSON формат:

```
{
  "EMPLOYEE_ID": 16,
  "ID": 1,
  "PRODUCT_ID": 19,
  "PROMOTION_ID": 6,
  "QUANTITY": 2,
  "SALE_DATE": "Thu, 13 Mar 2025 00:00:00 GMT",
  "VISITOR_ID": 17
}
```

### 3. ОПИСАНИЕ НА ENDPOINTS

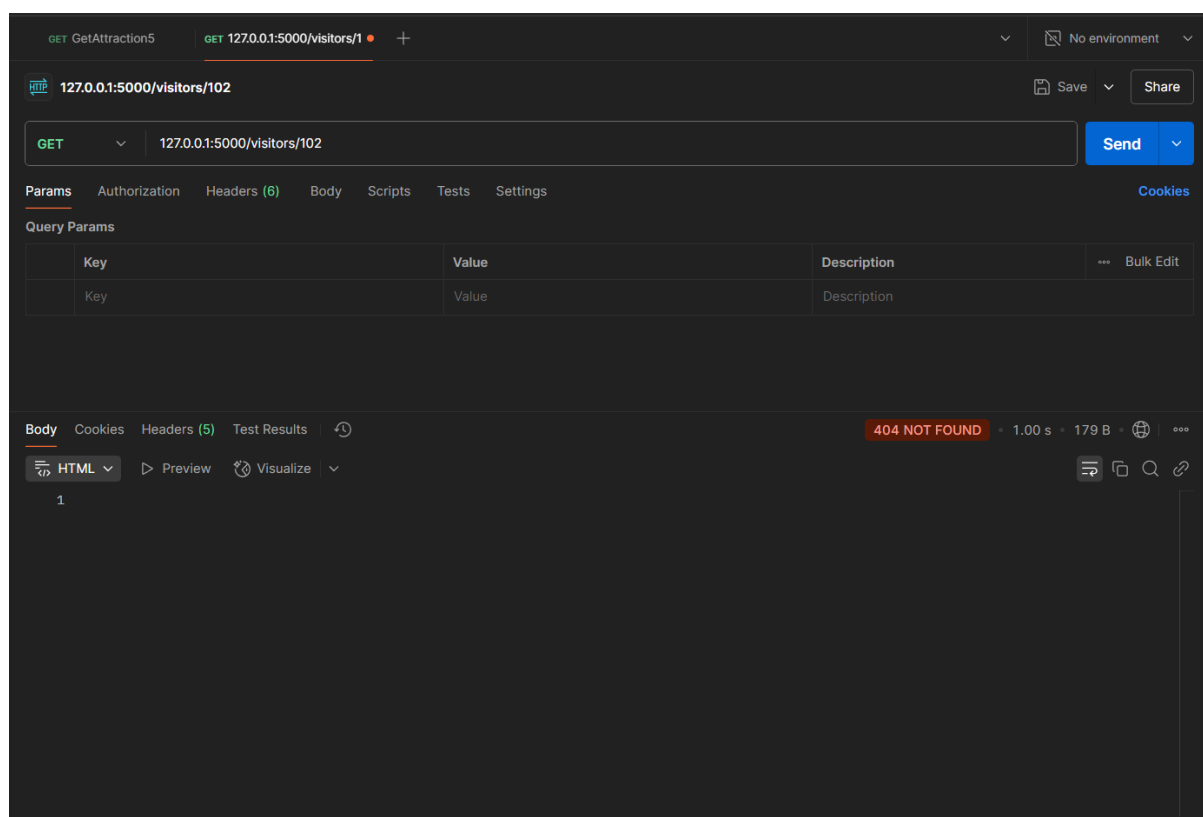
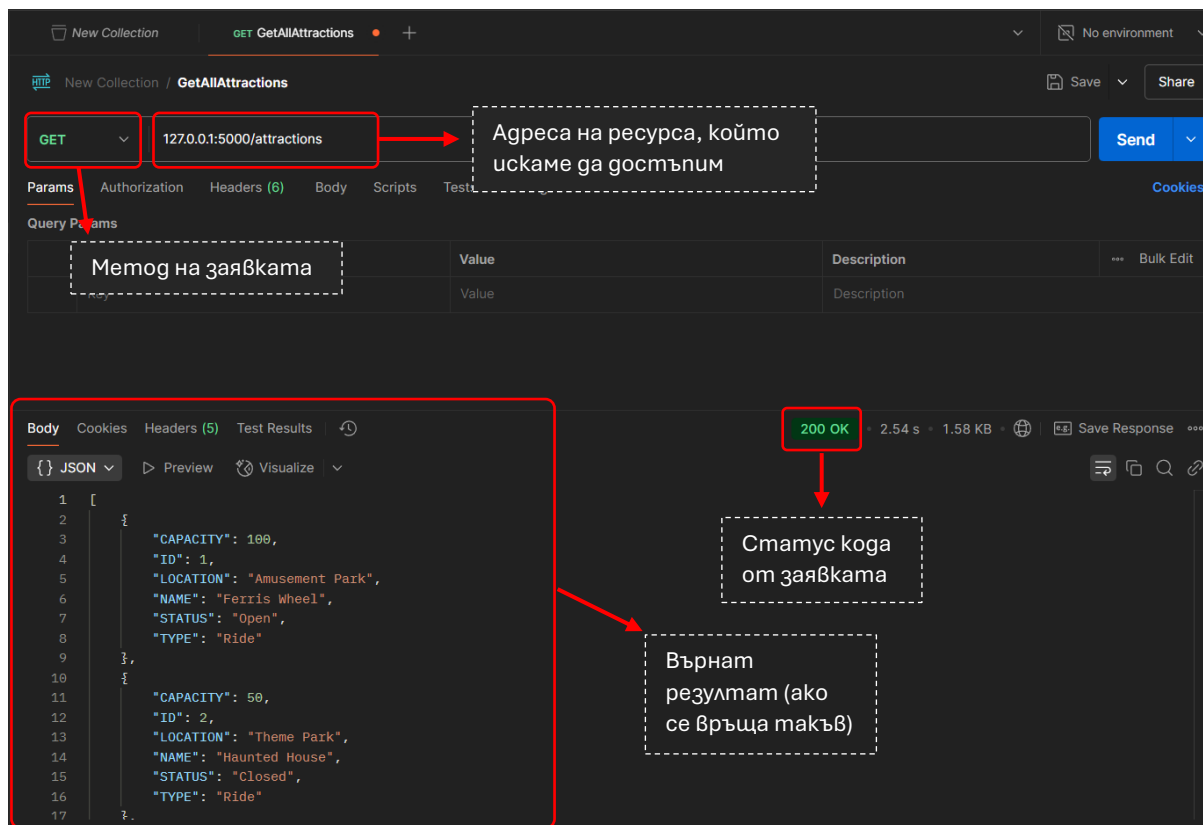
ТАБЛИЦА	МЕТОД	ОПИСАНИЕ	ФИЛТРАЦИЯ*	ВЪЗМОЖНИ ОТГОВОРИ	URL
Attractions	GET	Връща всички атракциони.	status, type	200, 400, 404	127.0.0.1:5000/attractions?status=Open&type=Water%20Ride
	GET	Връща <u>един</u> атракцион в зависимост от id-то му.	-	200, 400, 404	127.0.0.1:5000/attractions/5
Employees	GET	Връща всички служители.	job id, department id	200, 400, 404	127.0.0.1:5000/employees?job_id=5&dep_id=1
	GET	Връща <u>един</u> атракцион в зависимост от id-то му.	-	200, 400, 404	127.0.0.1:5000/employees/2
Events	GET	Връща всички събития.	type	200, 400, 404	127.0.0.1:5000/events?type=concert
	GET	Връща <u>едно</u> събитие в зависимост от id-то му.	-	200, 400, 404	127.0.0.1:5000/events/3
Sales	GET	Връща всички продажби.	visitor_id, product_id, employee_id	200, 400, 404	127.0.0.1:5000/sales
	GET	Връща <u>една</u> продажба в зависимост от id-то й.	-	200, 400, 404	127.0.0.1:5000/sales/5
Tickets	GET	Връща всички билети.	type, status	200, 400, 404	127.0.0.1:5000/tickets?type=Adult&status=available
	GET	Връща <u>един</u> билет в зависимост от id-то му.	-	200, 400, 404	127.0.0.1:5000/tickets/3
	POST	Добавя билет.	-	201, 500	127.0.0.1:5000/tickets

<b>Transactions</b>	GET	Връща всички транзакции.	type, payroll_id, maintenance_id, sale_id, ticket_id	200, 400, 404	127.0.0.1:5000/transactions
	GET	Връща една транзакция в зависимост от id-то ѝ.	-	200, 400, 404	127.0.0.1:5000/transactions/10
<b>Visitors</b>	GET	Връща всички посетители.	-	200, 400, 404	127.0.0.1:5000/visitors
	GET	Връща <u>един</u> посетител в зависимост от id-то му.	-	200, 400, 404	127.0.0.1:5000/visitors/8
	POST	Добавя посетител.	-	201, 500	127.0.0.1:5000/visitors
	PUT	Редактира <u>вече съществуващ</u> посетител.	-	200, 404, 500	127.0.0.1:5000/visitors/8
	DELETE	Изтрива <u>вече съществуващ</u> посетител.	-	200, 400, 404	127.0.0.1:5000/visitors/8
<b>Products</b>	GET	Връща всички продукти.	category	200, 400, 404	127.0.0.1:5000/products?category=food

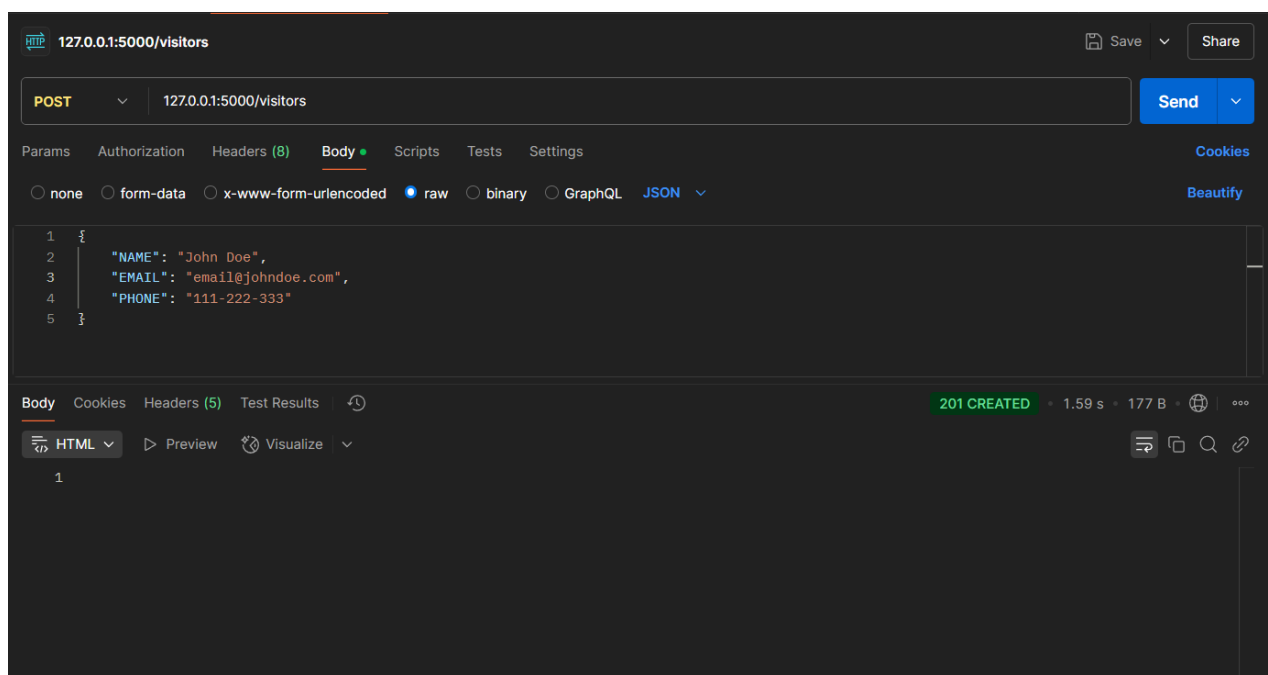
#### \*Филтрация:

Филтър може да се приложи с 1 един параметър или Всяка възможна комбинация от филтри, където има  $\geq 2$  параметъра. Пример – sales може да се филтрира само с employee\_id или visitor\_id и employee\_id едновременно, или product\_id и visitor\_id, и т.н.

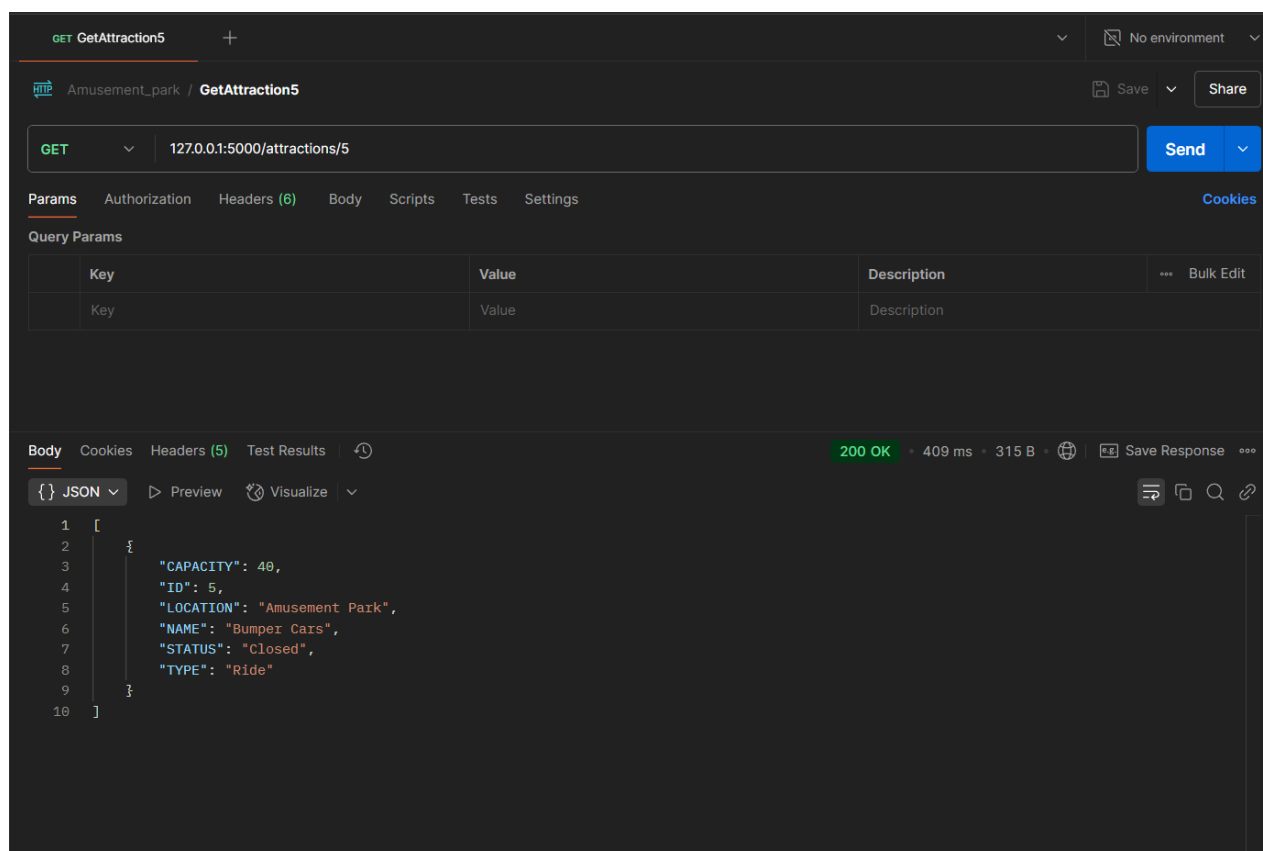
#### 4. POSTMAN ЗАЯВКИ (Тестване на системата)



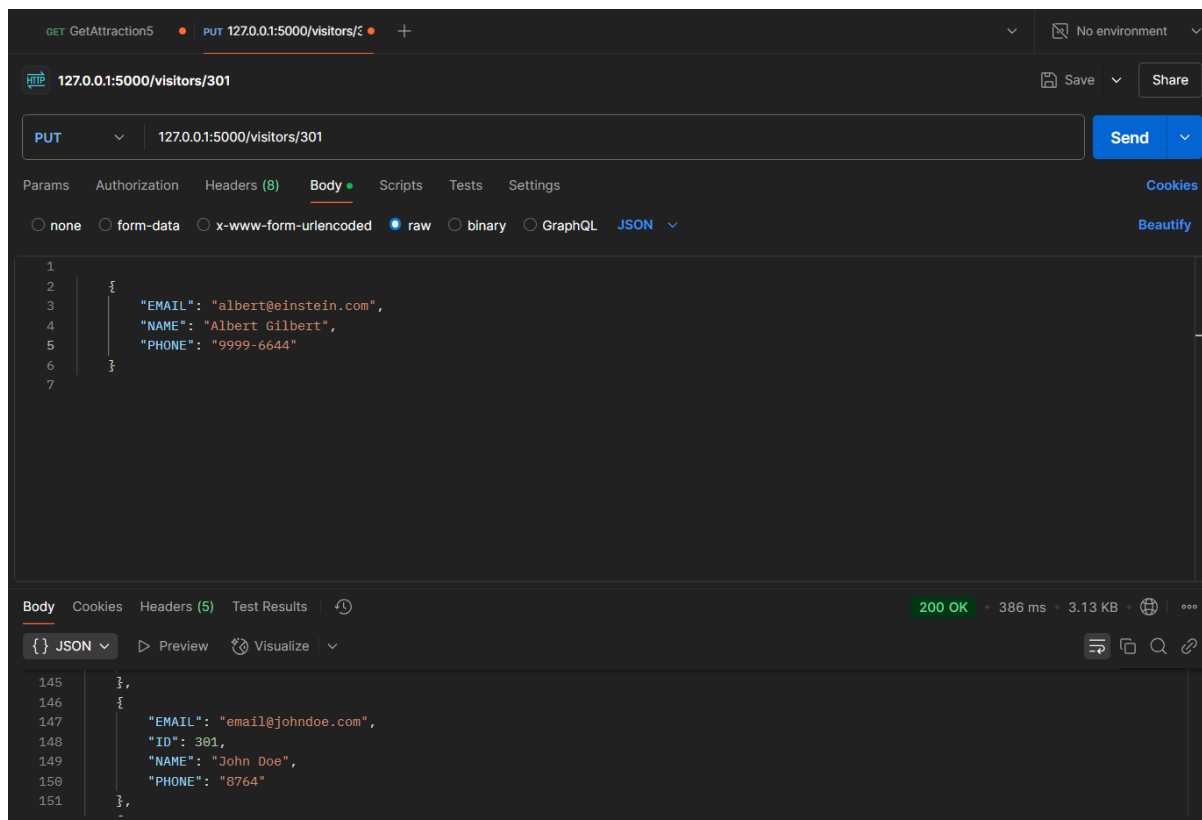
Фигура 1. Правим заявка за данните на посетител с ID 102. Такъв посетител няма! Сървърът връща код 404 – ресурсът не е намерен. Респективно – нямаме данни в JSON формат



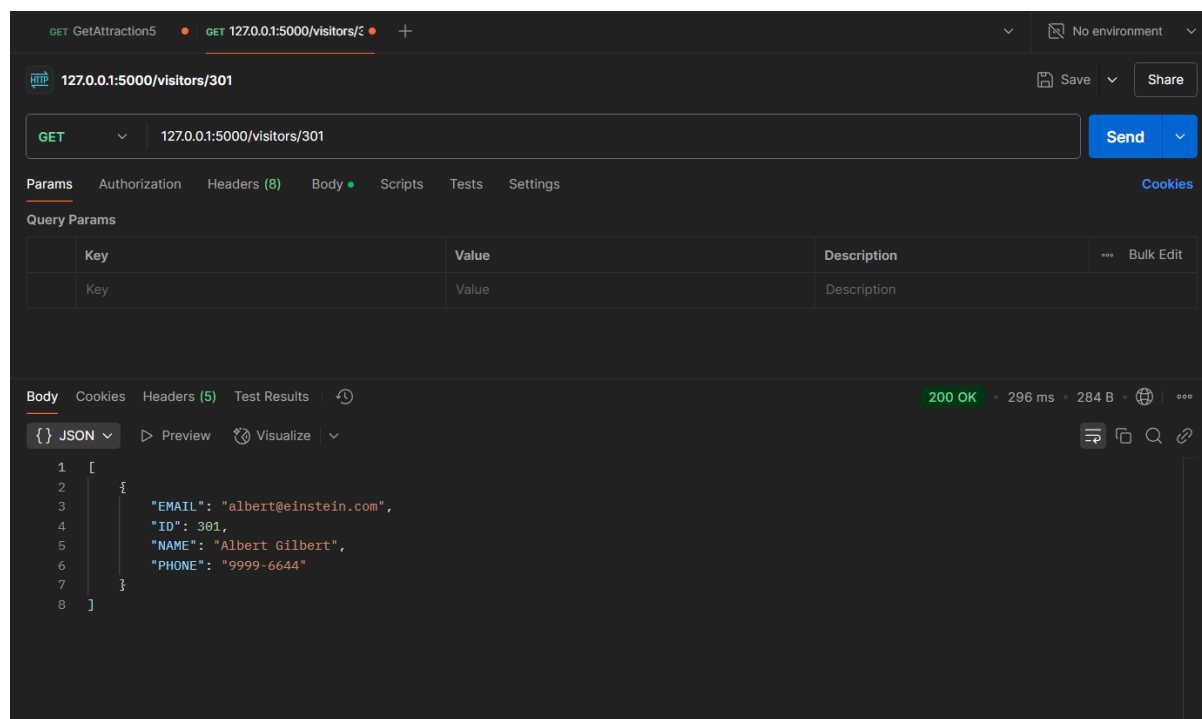
Фигура 2. POST заявка. Въвеждаме данните в JSON формат в секцията Body(raw, JSON). Сървърът връща код 201 (Успешно създаден) и текстово съобщение за успешно създаване на билет



Фигура 3. Правим заявка, която да ни извлече данните само за атракцион с ID 5. Връща се резултат – данните от таблицата; Сървърният отговор е 200 OK.



Фигура 4.1. PUT заявка. Въвеждаме данните в JSON формат в секцията Body(raw, JSON). В Body долу виждаме текущите данни за обекта в базата данни. Працаме заявката – връща 200 – записът е успешно редактиран.



Фигура 4.2. Правим GET заявка към посетителя, когото променихме. Сървърът връща 200 и данните на посетителя с ИД 301. Вижда се, че данните са променени и новите стойности са записани в базата.