

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:
 - 200 / OK – заявката е успешно обработена.
 - 201 / Created – заявката е успешна; като резултат е създаден нов ресурс.
 - 404 / Not Found – ресурсът не е намерен.
 - 500 / Internal Server Error – заявка, която не може да се обработи – грешка от сървърна страна.

2. ОПИСАНИЕ НА ENDPOINTS

URL: localhost:5000 +	МЕТОД	ОПИСАНИЕ	СЪРВЪРЕН ОТГОВОР
/attractions	GET	Извежда списък с всички атракциони	200 OK: JSON масив с атракциони <i>или</i> 404 Not Found: когато няма никакви записи на атракциони в базата
/attractions/<id> (/attractions/5)	GET	Извежда данни за атракцион, в зависимост от идентификатора му. В примера сървърът ще върне данни за атракцион с идентификатор 5.	200 OK: JSON обект с данни за атракционна <i>или</i> 404 Not Found: атракцион с този идентификатор не съществува в базата
/employees	GET	Извежда списък с всички служители	200 OK: JSON масив със служители <i>или</i> 404 Not Found: когато няма никакви записи на служители в базата

/employees/<id> (/employees/2)	GET	Извежда данни за служител, в зависимост от идентификатора му. В примера сървъра ще върне данни за служител с идентификатор 2.	200 OK: JSON обект с данни за служителя или 404 Not Found: служител с този идентификатор не съществува в базата
/sales	GET	Извежда списък с всички продажби	200 OK: JSON масив със продажби или 404 Not Found: когато няма никакви записи на продажби в базата
/sales/<id> (/sales/9	GET	Извежда данни за продажба, в зависимост от идентификатора му. В примера сървъра ще върне данни за продажба с идентификатор 9.	200 OK: JSON обект с данни за продажбата или 404 Not Found: продажба с този идентификатор не съществува в базата
/tickets	GET	Извежда списък с всички билети	200 OK: JSON масив със билети или 404 Not Found: когато няма никакви записи на билети в базата
/tickets/<id> (/tickets/3)	GET	Извежда данни за билет, в зависимост от идентификатора му. В примера сървъра ще върне данни за билет с идентификатор 3.	200 OK: JSON обект с данни за билет или 404 Not Found: билет с този идентификатор не съществува в базата
/tickets	POST	Създава нов билет и го запазва в базата. Очакваният формат на входните параметри е JSON и е следния: { "PRICE": float, "PROMOTION_ID": int, "STATUS": "string", "TYPE": "string", "VALIDITY_START": "YYYY-MM-DD", "VALIDITY_END": "YYYY-MM-DD" }	201 Created: Билетът е създаден или 500 Internal Server Error: билетът не е създаден, поради грешка

/events	GET	Извежда списък с всички специални събития.	200 OK: JSON масив със събития. или 404 Not Found: когато няма никакви записи на събития в базата
/events/<id> (/events/1)	GET	Извежда данни за събитие, в зависимост от идентификатора му. В примера сървъра ще върне данни за събитие с идентификатор 1.	200 OK: JSON обект с данни за събитие или 404 Not Found: събитие с този идентификатор не съществува в базата
/visitors	GET	Извежда списък с всички посетители.	200 OK: JSON масив с посетители. или 404 Not Found: когато няма никакви записи на посетители в базата
/visitors/<id> (/visitors/4)	GET	Извежда данни за посетител, в зависимост от идентификатора му. В примера сървъра ще върне данни за посетител с идентификатор 4.	200 OK: JSON обект с данни за посетител или 404 Not Found: посетител с този идентификатор не съществува в базата
/visitors	POST	Създава нов билет и го запазва в базата. Очакваният формат на входните параметри е JSON и е следния: { "NAME": "string", "EMAIL": "string", "PHONE": "string", }	201 Created: Посетителят е създаден или 500 Internal Server Error: посетителят не е създаден, поради грешка

/transactions	GET	Извежда списък с всички транзакции.	200 OK: JSON масив с транзакции. или 404 Not Found: когато няма никакви записи на посетители в базата
/transactions/<id> (/transactions/6)	GET	Извежда данни за транзакция, в зависимост от идентификатора му. В примера сървърът ще върне данни за транзакция с идентификатор 4.	200 OK: JSON обект с данни за транзакция или 404 Not Found: транзакция с този идентификатор не съществува в базата

3. POSTMAN ЗАЯВКИ (Тестване на системата)

The screenshot displays the Postman application interface for a GET request. At the top, the request method is set to 'GET' and the URL is '127.0.0.1:5000/attractions'. A red box highlights the URL, with an arrow pointing to a dashed box containing the text 'Адреса на ресурса, който искаме да достъпим'. Below the URL bar, the 'Params' tab is active, showing a table with two columns: 'Key' and 'Value'. A red box highlights the 'Key' column header, with an arrow pointing to a dashed box containing the text 'Метод на заявката'. The 'Body' tab is also visible, showing a JSON response. A red box highlights the JSON response, with an arrow pointing to a dashed box containing the text 'Върнат резултат (ако се връща такъв)'. The status bar at the top right shows '200 OK' in a green box, with an arrow pointing to a dashed box containing the text 'Статус кога от заявката'.

GET 127.0.0.1:5000/attractions

Адреса на ресурса, който искаме да достъпим

Params

Key	Value	Description

Метод на заявката

Body

```

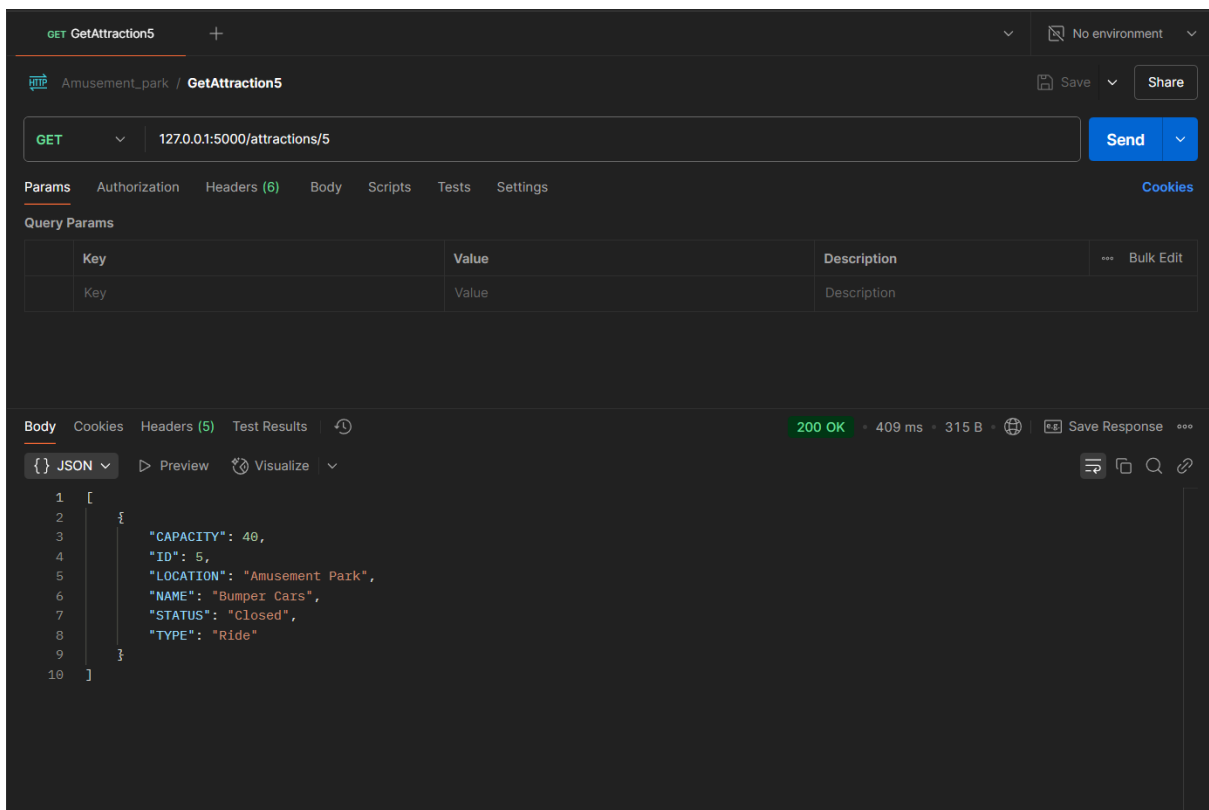
1 [
2   {
3     "CAPACITY": 100,
4     "ID": 1,
5     "LOCATION": "Amusement Park",
6     "NAME": "Ferris Wheel",
7     "STATUS": "Open",
8     "TYPE": "Ride"
9   },
10  {
11    "CAPACITY": 50,
12    "ID": 2,
13    "LOCATION": "Theme Park",
14    "NAME": "Haunted House",
15    "STATUS": "Closed",
16    "TYPE": "Ride"
17  }
18 ]

```

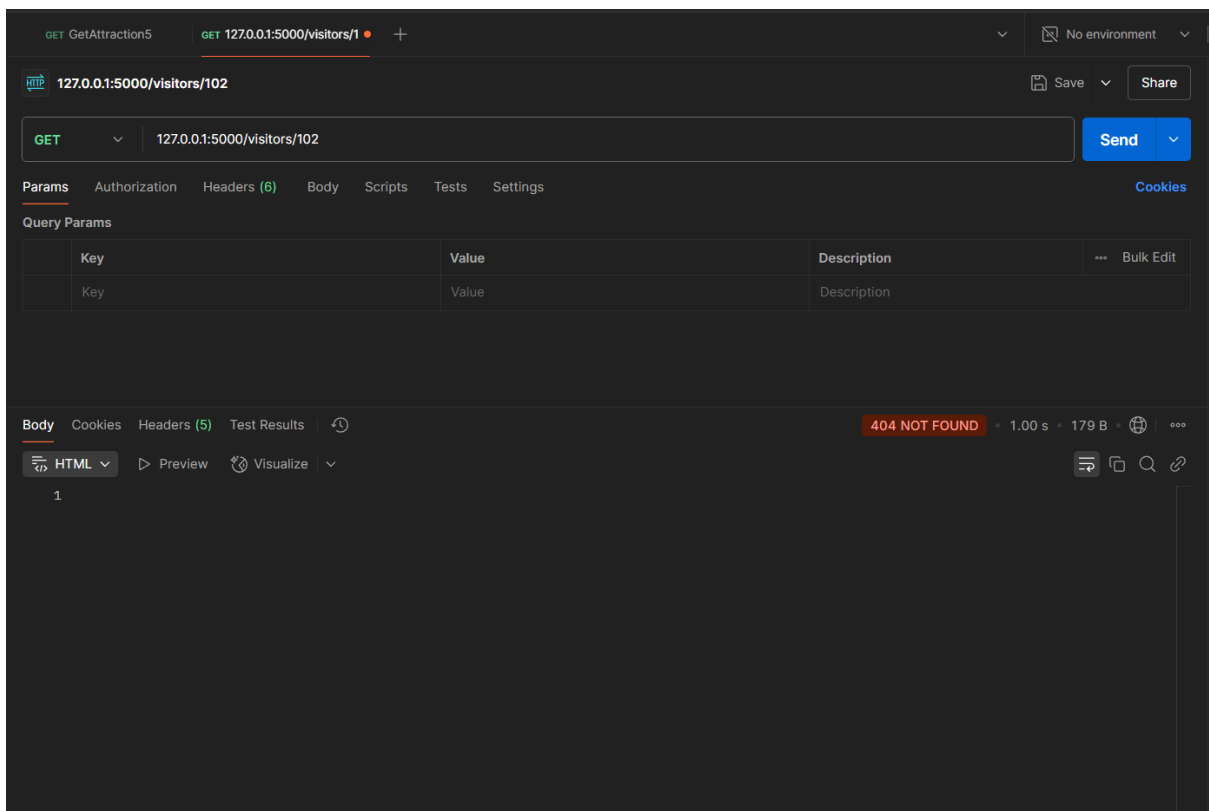
Върнат резултат (ако се връща такъв)

200 OK

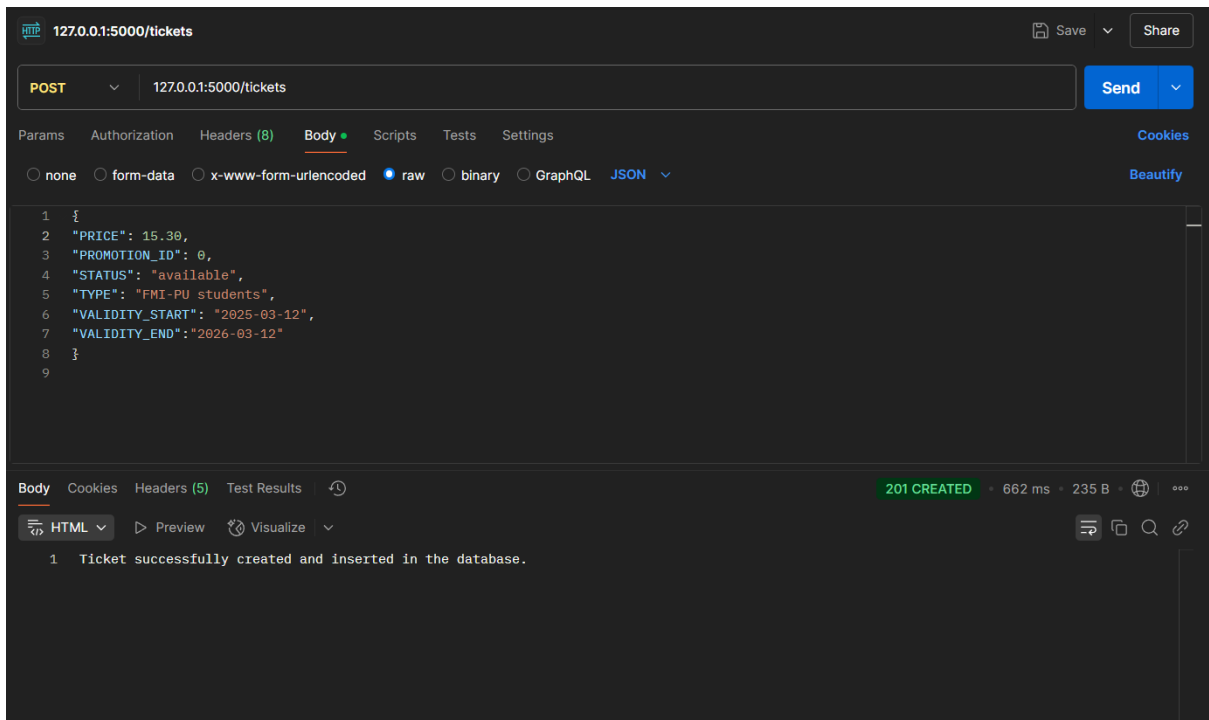
Статус кога от заявката



Фигура 2. Правим заявка, която да ни извлече данните само за атракцион с ID 5
Връща се резултат – данните от таблицата; Сървърния отговор е 200 OK.



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 3. 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

