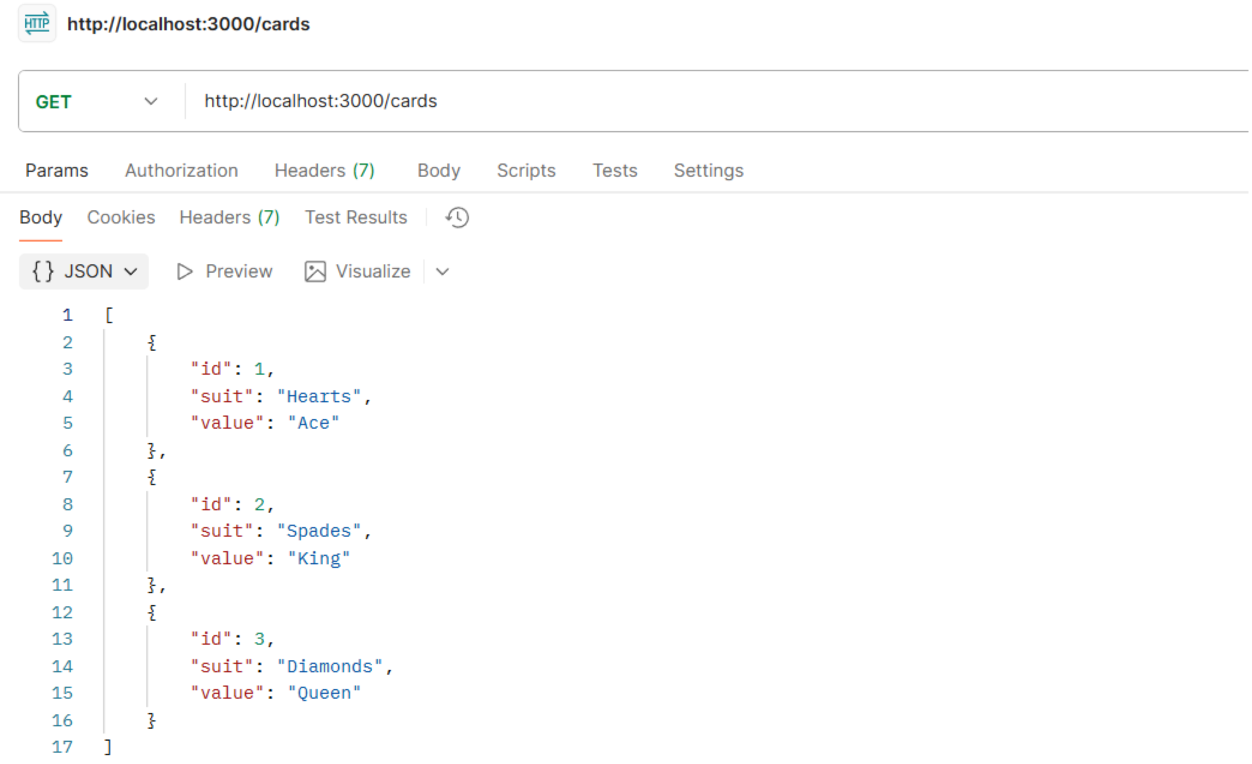
NAME	J k Kunal
UID	23BCS11041
CLASS	622-A

- NodeJs PRACTISE 2
- CODE

```
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

let cards = [
  { "id": 1, "suit": "Hearts", "value": "Ace" },
  { "id": 2, "suit": "Spades", "value": "King" },
  { "id": 3, "suit": "Diamonds", "value": "Queen" }
];

let nextId = cards.length + 1;

app.get('/cards', (req, res) => {
  res.status(200).json(cards);
});

app.get('/cards/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const card = cards.find(c => c.id === id);

  if (card) {
    res.status(200).json(card);
  } else {
    res.status(404).json({ message: `Card with ID ${id} not found` });
  }
});

app.post('/cards', (req, res) => {
  const { suit, value } = req.body;

  if (!suit || !value) {
    return res.status(400).json({ message: "Missing required fields: suit and value" });
  }

  const newCard = {
    id: nextId++,
    suit,
    value
  };

  cards.push(newCard);

  res.status(201).json(newCard);
});

app.delete('/cards/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const cardIndex = cards.findIndex(c => c.id === id);

  if (cardIndex !== -1) {
    const [deletedCard] = cards.splice(cardIndex, 1);

    res.status(200).json({
      message: `Card with ID ${id} removed`,
      card: deletedCard
    });
  } else {
    res.status(404).json({ message: `Card with ID ${id} not found` });
  }
});

app.listen(port, () => {
  console.log(`Card API listening at http://localhost:${port}`);
});
```

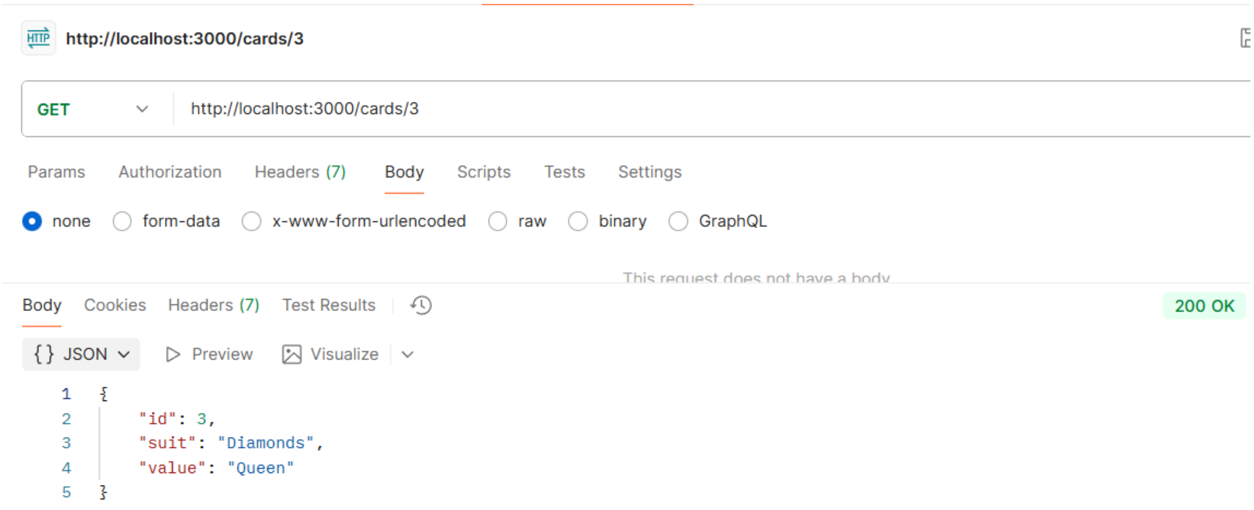
- OUTPUT



http://localhost:3000/cards

GET http://localhost:3000/cards

```
[
    {
        "id": 1,
        "suit": "Hearts",
        "value": "Ace"
    },
    {
        "id": 2,
        "suit": "Spades",
        "value": "King"
    },
    {
        "id": 3,
        "suit": "Diamonds",
        "value": "Queen"
    }
]
```



http://localhost:3000/cards/3

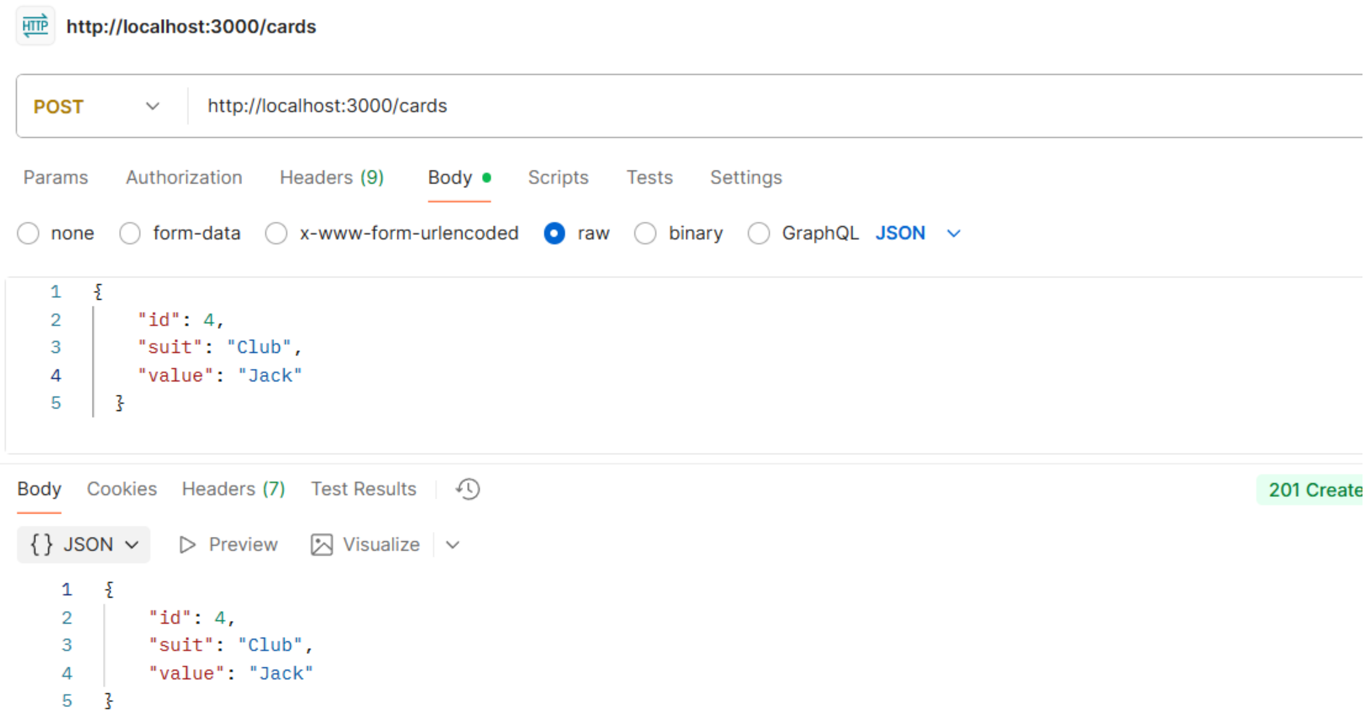
GET http://localhost:3000/cards/3

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

200 OK

```
{
    "id": 3,
    "suit": "Diamonds",
    "value": "Queen"
}
```



http://localhost:3000/cards

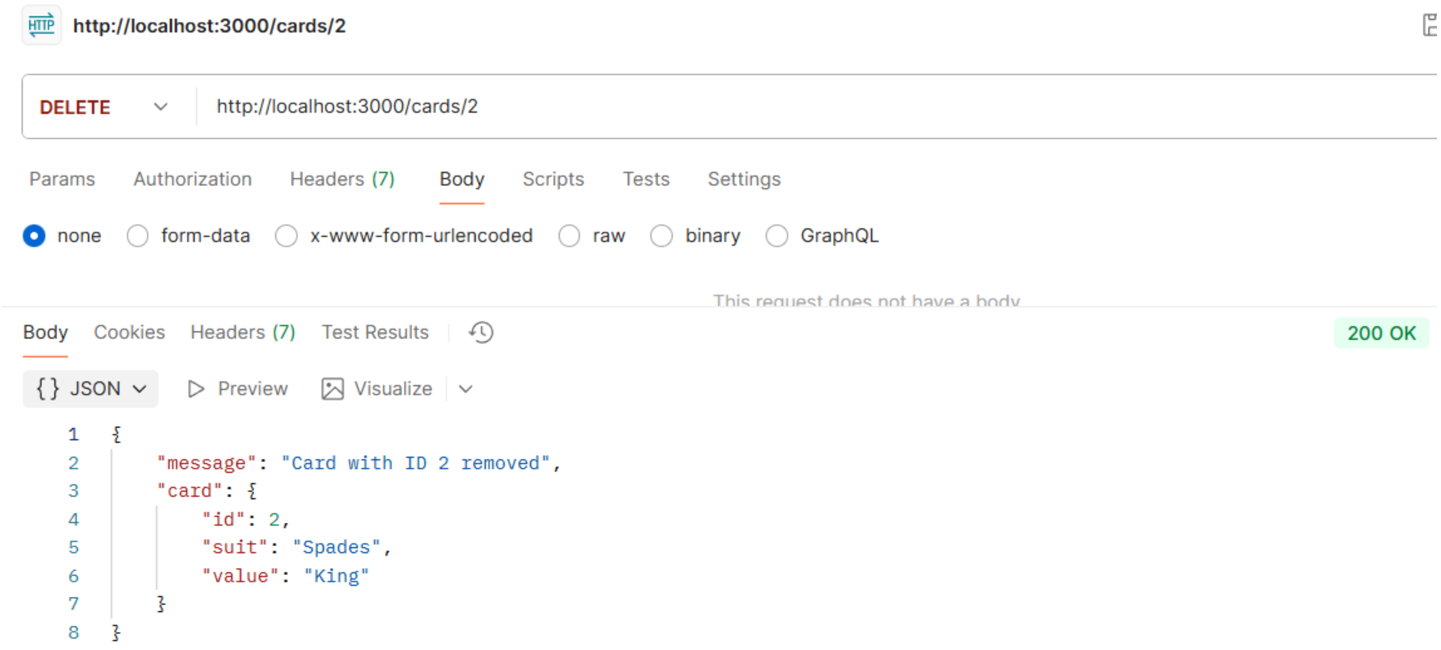
POST http://localhost:3000/cards

Body ● — ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id": 4,
    "suit": "Club",
    "value": "Jack"
}
```

Body — {} JSON — 201 Create

```
{
    "id": 4,
    "suit": "Club",
    "value": "Jack"
}
```



http://localhost:3000/cards/2

DELETE http://localhost:3000/cards/2

Body — ● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Body — {} JSON — 200 OK

```
{
    "message": "Card with ID 2 removed",
    "card": {
        "id": 2,
        "suit": "Spades",
        "value": "King"
    }
}
```

Output Link :- http://localhost:3000/cards

API’s: GET http://localhost:3000/cards GET http://localhost:3000/cards/3

POST http://localhost:3000/cards DELETE http://localhost:3000/cards/2