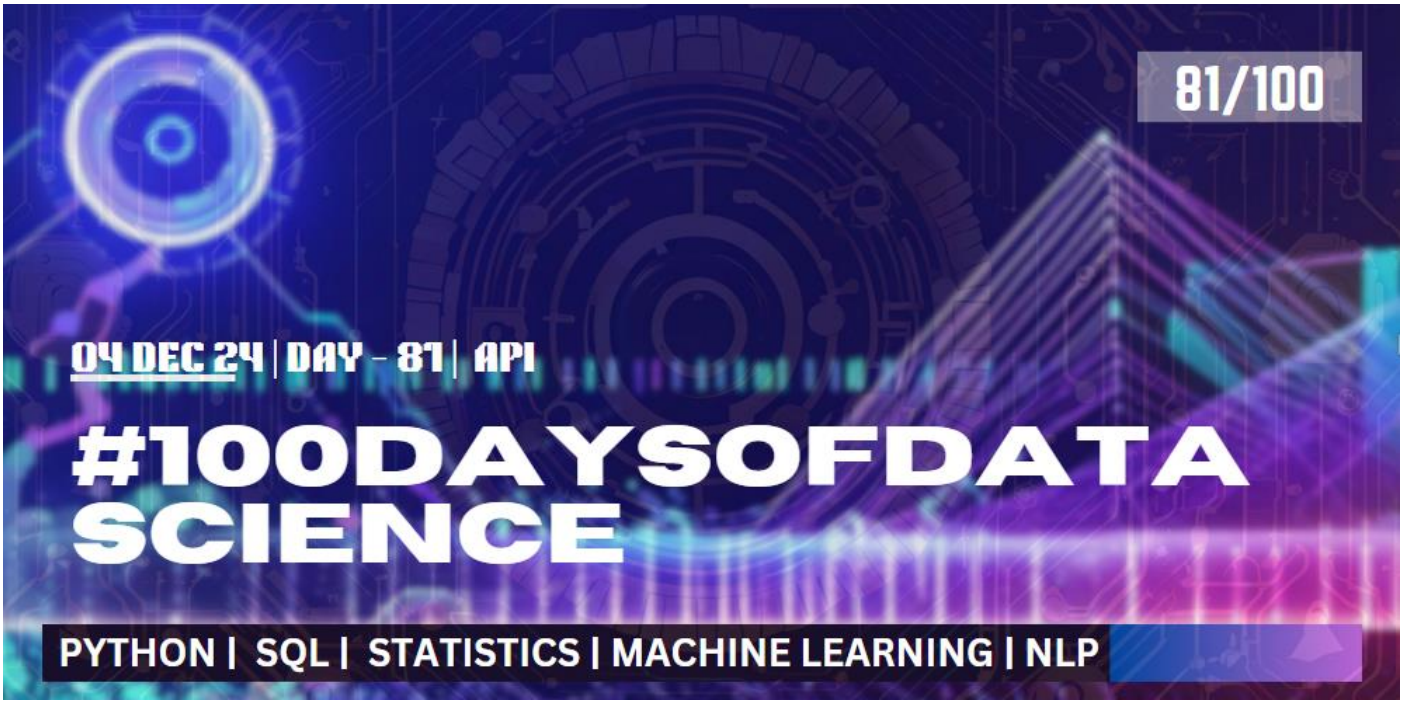


API (Application Programming Interfaces)

What are APIs?

APIs (Application Programming Interfaces) are intermediaries that enable two software applications to communicate with each other. APIs define rules and protocols that allow developers to access specific features or data of an application, service, or platform, often abstracting the complexity of backend operations.

Key Features of APIs

1. **Standardized Communication**
   APIs provide a standardized way for different systems to interact, making integration between platforms seamless and efficient.
2. **Data Access and Manipulation**
   APIs enable the retrieval, creation, updating, or deletion of data from external systems without requiring direct database access.
3. **Extensibility**
   They allow developers to extend the functionality of applications by integrating external services or adding new features.
4. **Scalability and Efficiency**
   APIs are designed to handle multiple requests efficiently, ensuring scalability for large-scale applications.

Applications of APIs

- **Social Media Integration**: Fetch posts, user data, or trends from platforms like Twitter, Instagram, or Facebook.
- **Payment Processing**: Use APIs like Stripe or PayPal to enable secure transactions.
- **Weather Updates**: Fetch real-time weather data for integration into apps or websites.
- **Third-party Authentication**: Implement OAuth with APIs like Google, Facebook, or LinkedIn for login functionality.

Best Practices for Working with APIs

- **Read Documentation**: Always consult API documentation to understand endpoints, request formats, and authentication requirements.

- **Handle Errors Gracefully**: Implement error handling to manage failed requests or unexpected responses.
- **Rate Limiting**: Respect API rate limits to avoid throttling or being blocked.
- **Secure API Keys**: Store API keys securely and avoid exposing them in public repositories.

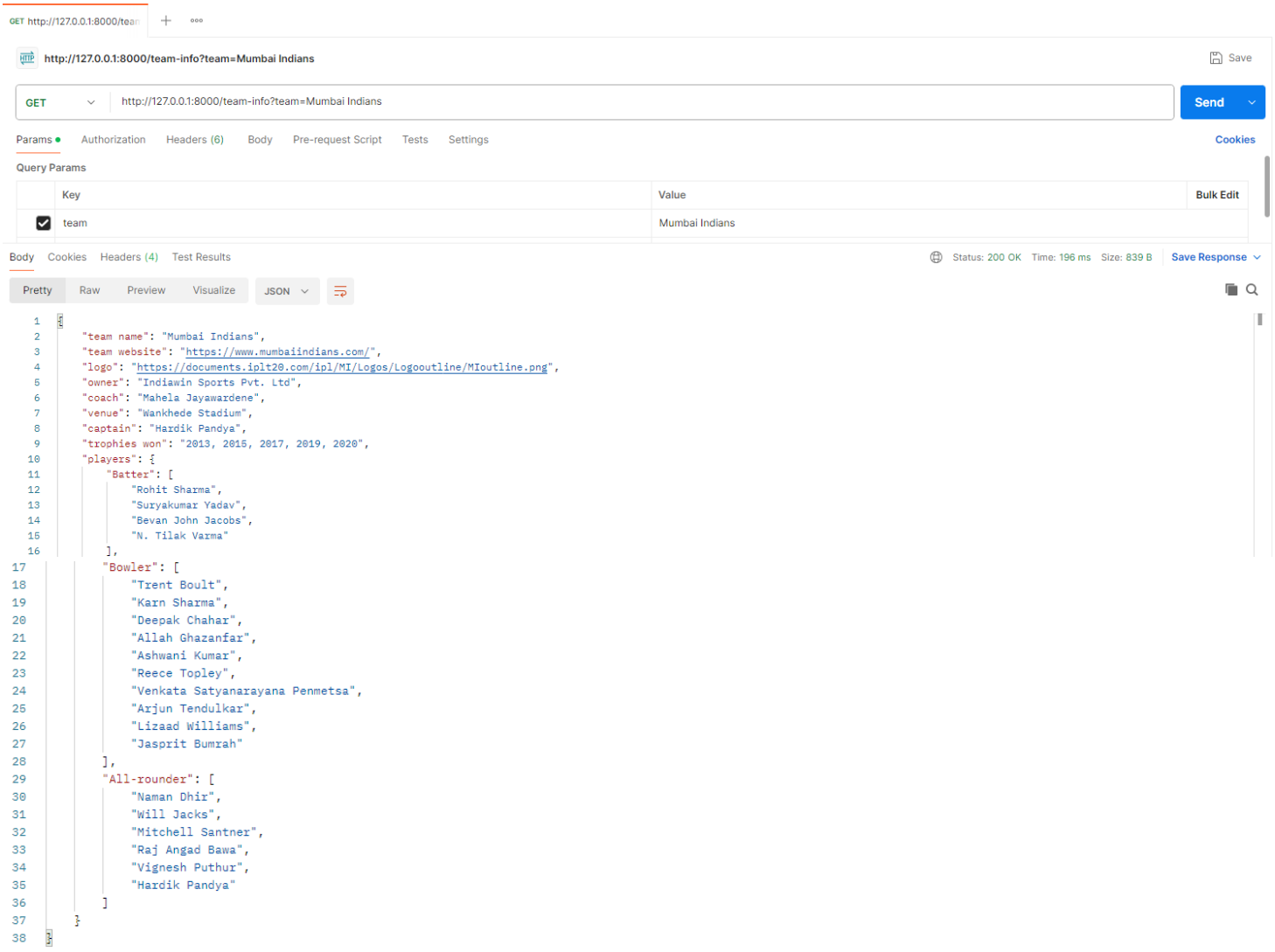
Commonly Used Methods in APIs

1. **GET**: Retrieve data from a server.
   Example: Fetching user details from a database.
2. **POST**: Send data to the server to create a new resource.
   Example: Submitting a form to register a new user.
3. **PUT**: Update an existing resource on the server.
   Example: Modifying a user's profile information.
4. **DELETE**: Remove a resource from the server.
   Example: Deleting a user's account.

Common Libraries for Working with APIs

1. **Requests (Python)**: A simple and intuitive library for making HTTP requests.
2. **Postman**: A GUI tool to test, debug, and document APIs effectively.

Example: Using FastAPI and APIs
Using Postman



Using Python

team_details_fetch.py > ...

```python
import requests

team = 'Mumbai Indians'


response = requests.get(f"http://127.0.0.1:8000/team-info?team={team}")
output = response.json()
print(output)
```

```
C:\Users\Zahid.Shaikh\100days\80>python team_details_fetch.py
{'team name': 'Chennai Super Kings', 'team website': 'https://www.chennaisuperkings.com/', 'logo': 'https://documents.iplt20.com/ipl/CSK/logos/Logooutline
/CSKoutline.png', 'owner': 'Chennai Super Kings Cricket Ltd.', 'coach': 'Stephen Fleming', 'venue': 'M. A. Chidambaram Stadium', 'trophies won': '2010, 20
11, 2018, 2021, 2023', 'players': {'Batter': ['Ruturaj Gaikwad', 'Devon Conway', 'Rahul Tripathi', 'Shaik Rasheed', 'Andre Siddarth'], 'Bowler': ['Khaleel
 Ahmed', 'Noor Ahmad', 'Mukesh Choudhary', 'Gurjapneet Singh', 'Nathan Ellis', 'Shreyas Gopal', 'Matheesha Pathirana'], 'All-rounder': ['Rachin Ravindra',
 'Ravichandran Ashwin', 'Vijay Shankar', 'Sam Curran', 'Anshul Kamboj', 'Deepak Hooda', 'Jamie Overton', 'Kamlesh Nagarkoti', 'Ramakrishna Ghosh', 'Ravind
ra Jadeja', 'Shivam Dube']}}
```

This example demonstrates how to create a ipl-team-data-fetching endpoint using FastAPI. **FastAPI** ensures rapid development and seamless API handling, making it a robust choice for building APIs. 🚀