

API TESTING BASICS

WHAT TO EXPECT



- 1 API & UI Testing
- 2 XML & JSON
- 3 API Testing Construct
- 4 HTTP Methods & Meaning
- 5 HTTP Response Codes

If you thought -

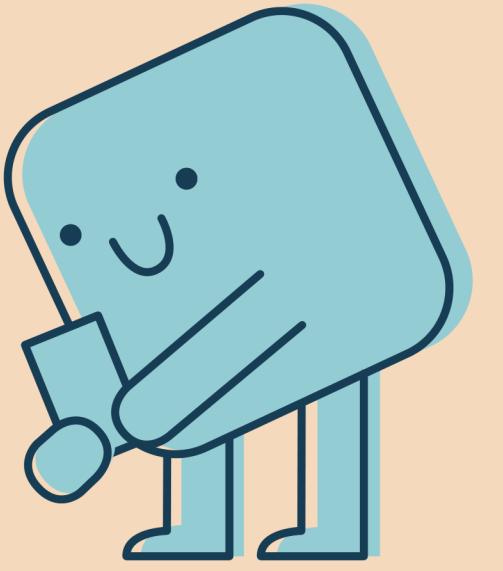
API Testing is going to involve some nice UI, Images, Graphics to test



API testing is very different to UI Testing

- API testing primarily deals with the communication between software components at the code level, focusing on data exchange and functionality.
- API testing verifies the functionality and performance of the back-end processes, ensuring that data is correctly processed and exchanged.
- API testing involves validating data exchange formats such as JSON or XML and ensuring proper communication between different software components.
- In contrast, UI testing is concerned with the visual and interactive aspects of the user interface.
- UI testing, on the other hand, focuses on the end-user experience, including layout, design, and user interactions.
- UI testing, in contrast, is concerned with how information is presented and interacted with by end-users.





**Lets review 2 components which you will
prominently come across in API Testing**

XML

Elements

Data

XML Declaration

```
xml
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="fiction">
    <title>Harry Potter</title>
    <author>J.K. Rowling</author>
    <price>29.99</price>
  </book>
  <book category="non-fiction">
    <title>Introduction to XML</title>
    <author>John Doe</author>
    <price>19.95</price>
  </book>
</bookstore>
```

What is XML ?

Extensible Markup Language (XML):

XML is a versatile and extensible markup language designed for storing and transporting data in a structured format.

Hierarchical Structure:

XML documents have a hierarchical structure consisting of elements, which are enclosed within start and end tags. This tree-like structure allows for the representation of relationships between different data elements.

Human-Readable and Machine-Readable:

XML is both human-readable and machine-readable, making it easy for developers to create and understand, as well as for software systems to parse and process.

Platform-Independent:

XML is platform-independent, meaning that it can be used across different operating systems and architectures. This makes it a widely adopted standard for data interchange in diverse computing environments.

Use Cases:

XML is commonly used for a variety of purposes, including data storage, configuration files, web services communication, and data exchange between disparate systems. Its flexibility and adaptability make it suitable for representing a wide range of data structures and content.

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="fiction">
    <title>Harry Potter</title>
    <author>J.K. Rowling</author>
    <price>29.99</price>
  </book>
  <book category="non-fiction">
    <title>Introduction to XML</title>
    <author>John Doe</author>
    <price>19.95</price>
  </book>
</bookstore>
```

JSON

key:value pair



json

```
{  
  "name": "John Doe",  
  "age": 30,
```

key:value pair



```
  "isStudent": false,  
  "grades": [95, 88, 72],  
  "address": {
```

represents an array

key:value pair



```
    "city": "New York",  
    "zipcode": "10001"
```

},

"courses": [

{

```
    "courseName": "Mathematics",  
    "instructor": "Professor Smith"
```

},

{

```
    "courseName": "Computer Science",  
    "instructor": "Dr. Johnson"
```

}

]

}

represents an nested object in main object

What is JSON ?

JavaScript Object Notation (JSON):

JSON is a lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate.

Key-Value Pairs:

JSON data is organized as key-value pairs, where each key is a string and each value can be a string, number, boolean, null, object, or array. This structure allows for the representation of complex data structures.

Human-Readable and Lightweight:

JSON is designed to be both human-readable and lightweight. Its syntax is simple, and it is easy for developers to understand and work with. This makes it a popular choice for data exchange in web development.

Data Structures:

JSON supports common data structures such as objects and arrays, allowing for the creation of hierarchical and nested structures to represent various forms of data.

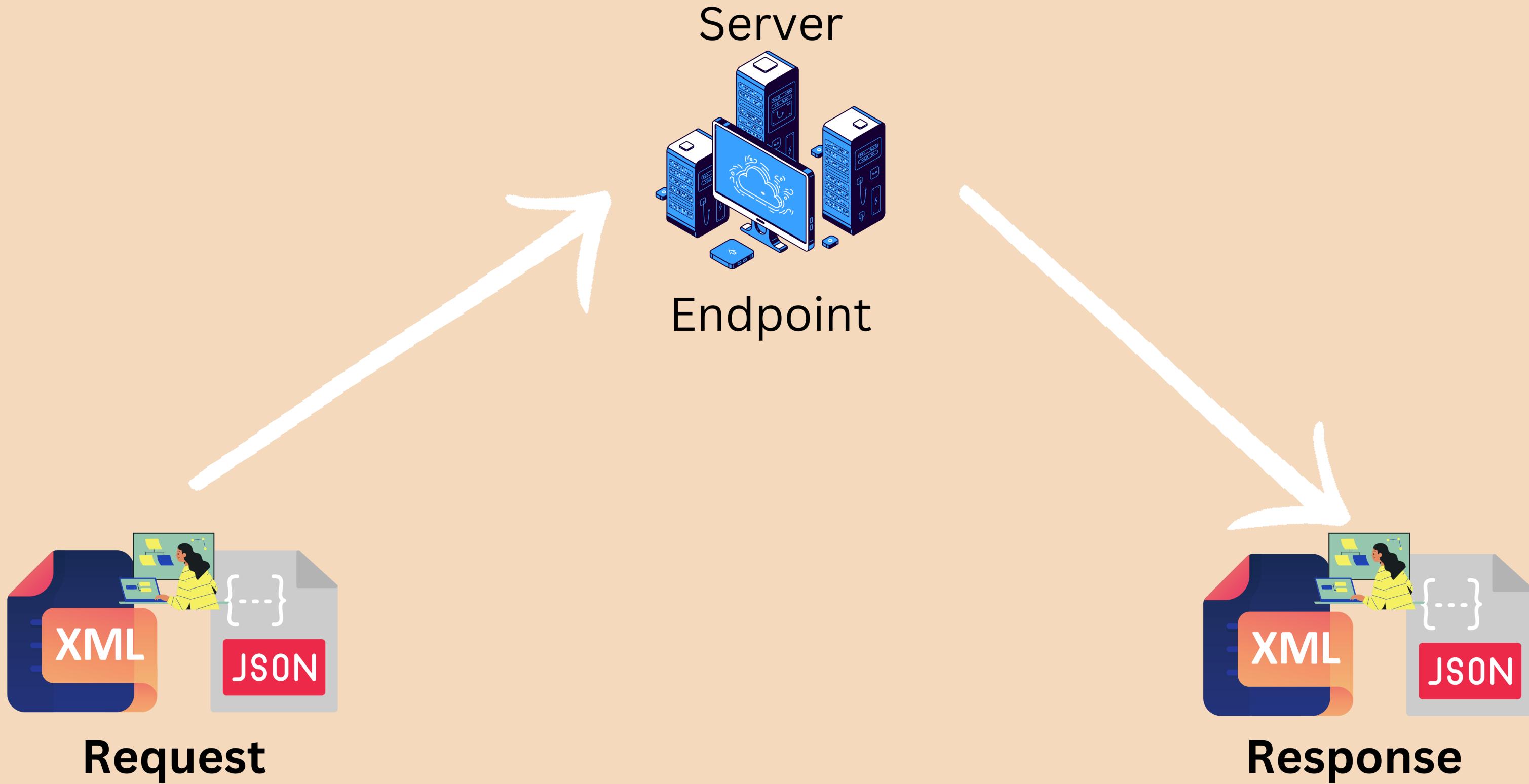
Language Independence:

While inspired by JavaScript object literal syntax, JSON is language-independent. It is widely supported in various programming languages, making it a versatile choice for data interchange between different systems and platforms.

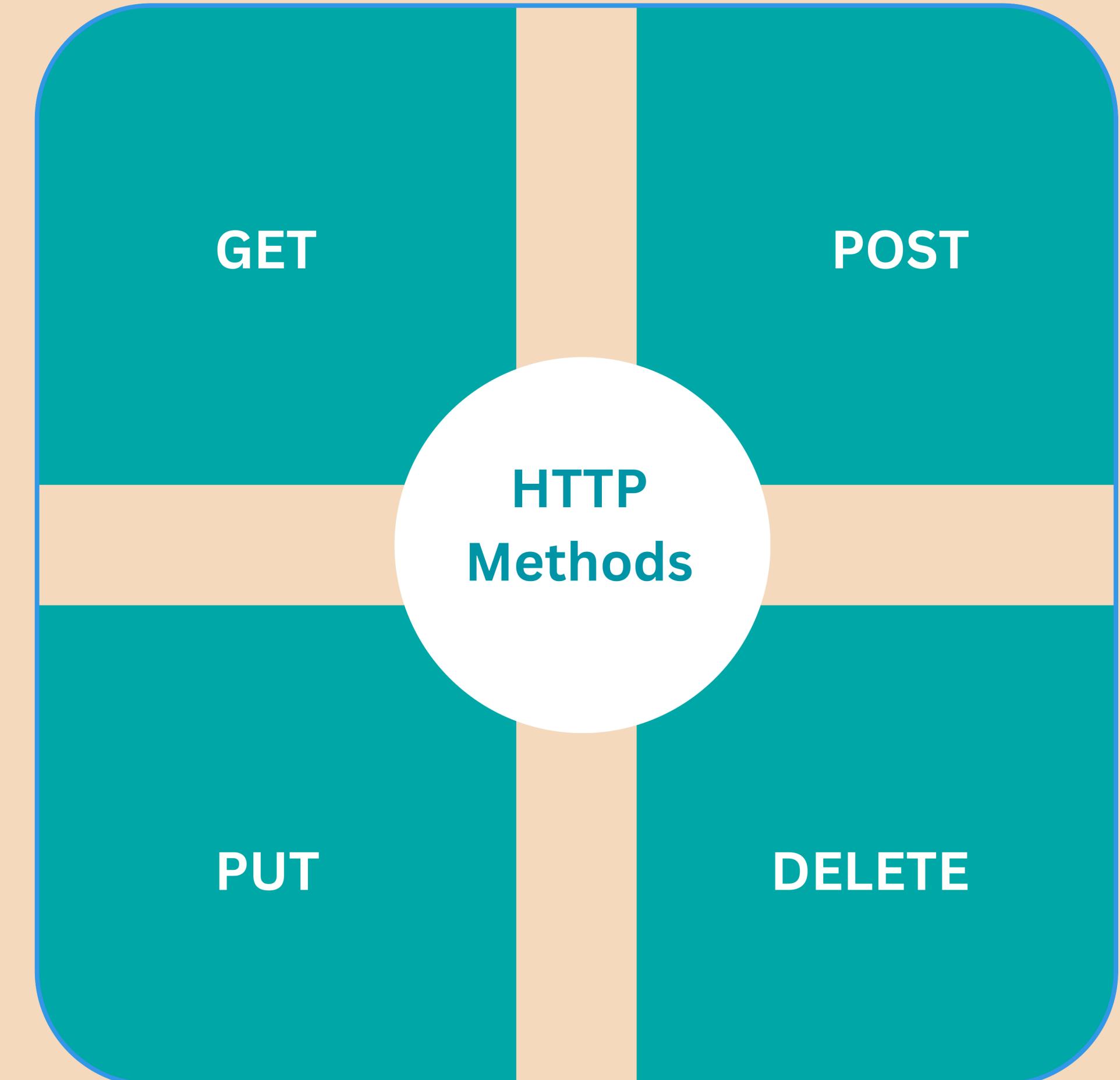
json

```
{  
  "name": "John Doe",  
  "age": 30,  
  "isStudent": false,  
  "grades": [95, 88, 72],  
  "address": {  
    "city": "New York",  
    "zipcode": "10001"  
  },  
  "courses": [  
    {  
      "courseName": "Mathematics",  
      "instructor": "Professor Smith"  
    },  
    {  
      "courseName": "Computer Science",  
      "instructor": "Dr. Johnson"  
    }  
  ]  
}
```

Construct of API Testing



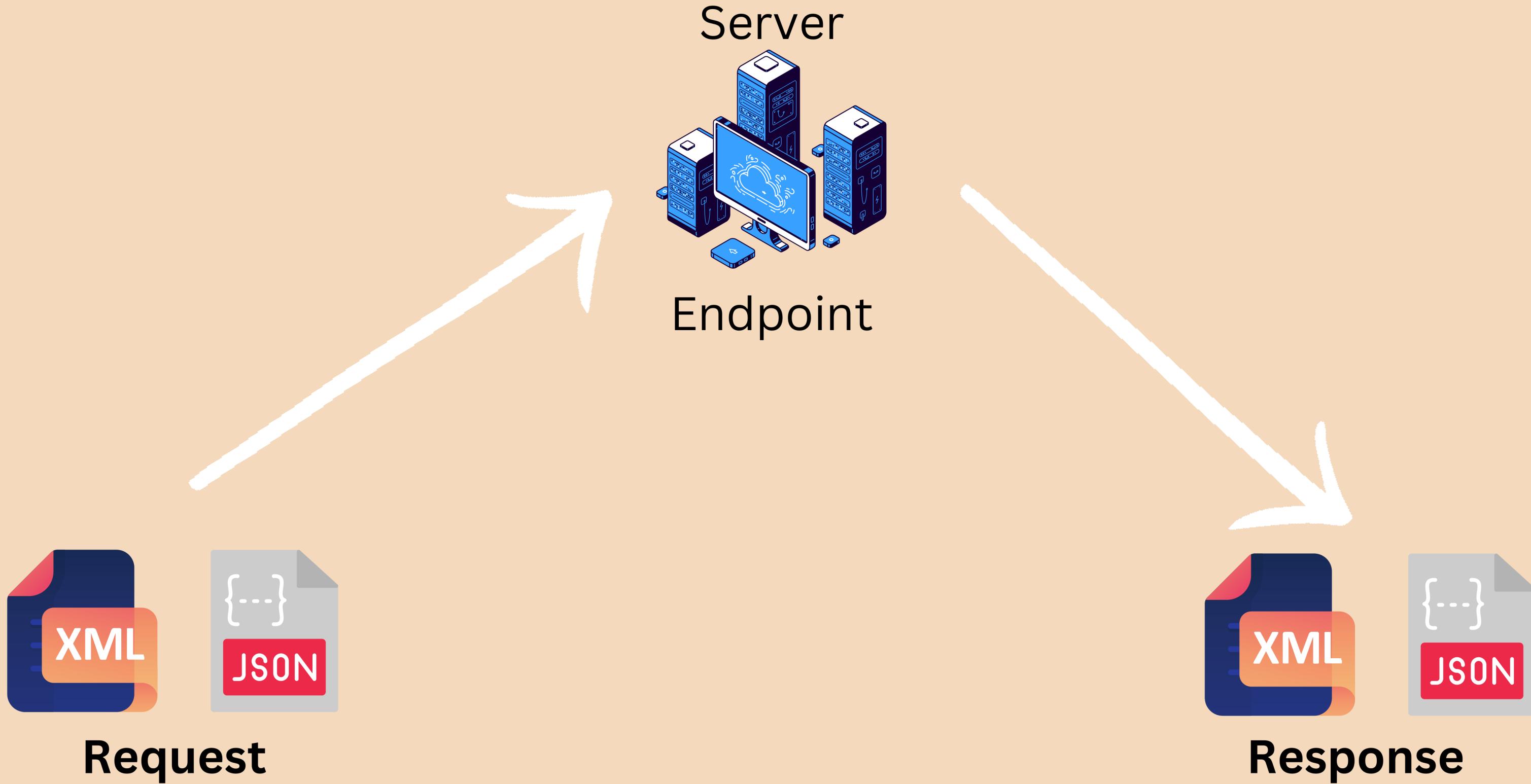
HTTP Methods



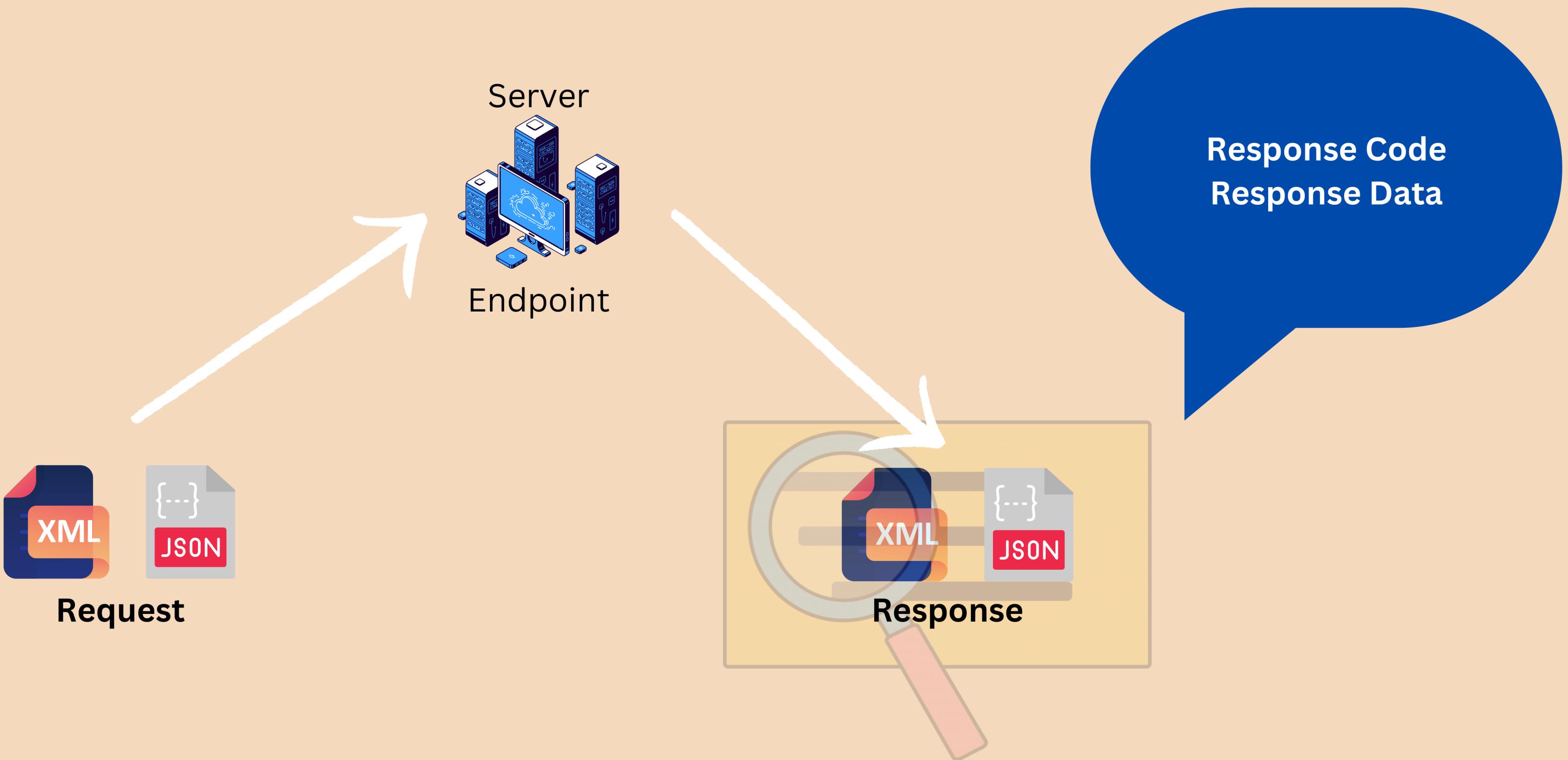
HTTP Methods Definition

Request Methods	Meaning	
POST	Insert / <u>Create</u> Data	C
GET	<u>Read</u> Data	R
PUT or PATCH	<u>Update</u> Data	U
DELETE	<u>Delete</u> Data	D

Construct of API Testing



Construct of API Testing



HTTP Response Codes

1XX

Informational

2XX

Success

3XX

Redirection

4XX

Client Error

5XX

Server Error



“

See you in Next Video

