

JSON Handling in Flask REST APIs (Developer Documentation)

1. Introduction

JSON (JavaScript Object Notation) is a lightweight data-interchange format widely used in RESTful APIs for communication between clients and servers.

2. JSON Basics

JSON consists of key–value pairs, arrays, and nested objects.

Example JSON:

```
{  
  "name": "Gowtham",  
  "email": "gowtham@example.com",  
  "role": "Backend Developer"  
}
```

3. Receiving JSON Data in Flask

Use `request.get_json()` to parse incoming JSON payloads.

4. Sending JSON Responses

Use `jsonify()` to safely return JSON responses.

5. JSON Validation and Error Handling

Always validate JSON fields and handle invalid JSON errors.

6. Common JSON Handling Errors

- Invalid JSON format
- Missing fields
- Wrong content-type
- Empty request body

7. JSON Handling with MongoDB

JSON data is automatically converted to BSON when inserted using PyMongo.

8. Best Practices

- Validate all JSON input
- Use correct HTTP status codes
- Log invalid requests