



Hands-On GenAI: LLMs, RAGs, and Agentic Systems for Beginners

Day 4

Adya Bhat and Tejas Venugopalan



Components of the web API

1. HTTP methods
2. Request schema: JSON body + query parameters
3. Response schema: JSON + status code + headers
4. Endpoint/ URL path: where the resource can be accessed (inserted/ updated/ deleted, etc.) e.g. `localhost:5000/movie`
5. API Semantics: a POST method, for the endpoint `localhost:5000/movie` should add a new entry in the data in the server/ backend/ database

Writing Flask Code- a template

Importing Flask, Flask-CORS libraries

```
from flask import Flask, jsonify, request  
from flask_cors import CORS
```

```
app = Flask(__name__)
```

Creates the Flask app, using the name of the current file (module).

```
CORS(app)
```

Allows cross-origin requests (from frontend for e.g.) for the app (backend), by adding headers to the request, for the specific methods.

Writing Flask Code- a template

Initializing data in the server/ creation of a connection to a database

```
db = [{"Movie": "Baasha"}, {"Movie": "Guntur Karraam"},  
{"Movie": "KGF"}]
```

Initializes data in the server file itself. For external databases, a connection must be established between the server and the database.

Writing Flask Code- a template

Initializing data in the server/ creation of a connection to a database

```
@app.route(<path>, methods=<list of methods>)
```

```
def what_the_function_does():
```

```
    # code to extract data from request and server/ database
```

```
    # code to process the data
```

```
    return jsonify(<message or data dictionary>), <request status  
code>
```

Python decorator extends functionality by mapping path to the corresponding function defined below it.

Writing Flask Code- a template

Running the app

```
if __name__ == "__main__":  
    app.run(debug=True)
```

Running the app

HTTP requests

Method	Purpose	Example
GET	Retrieve data from the server (read-only).	Fetch a list of all products from an API.
POST	Create a new resource on the server.	Add a new product to the inventory.
PUT	Update an existing resource entirely.	Update product details (overwrite all old data).
DELETE	Remove a resource from the server.	Delete a product from the inventory.



HTTP requests

Status Code	Meaning / Use Case
200 OK	Request was successful.
201 Created	A new resource was created successfully.
400 Bad Request	The server couldn't understand the request due to invalid syntax or missing data.
401 Unauthorized	Authentication required or invalid credentials provided.
403 Forbidden	The user is authenticated but doesn't have permission to access the resource.



HTTP requests

Status Code	Meaning / Use Case
404 Not Found	The requested resource doesn't exist on the server.
409 Conflict	The request could not be completed due to a conflict with the current state of the resource.
500 Internal Server Error	A generic server error occurred due to an unexpected condition.
502 Bad Gateway	The server received an invalid response from another server.
503 Service Unavailable	The server is temporarily unable to handle the request (maintenance or overload).

