Bakerina Swan Lake

APIs and AI-Driven Development

Hello!

Anjana Supun

anjanas@wso2.com | Associate Technical Lead | @ballerinalang | WSO2

Dulmina Kodagoda

dulmina@wso2.com | Senior Software Engineer | @ballerinalang | WSO2



About this Session





1. API Fundamentals

2. What are REST APIs

3. What is GraphQL

4. REST vs GraphQL

5. Microservices

6. WebSockets: Hands-on Session

7. The Rewards Challenge



What is an API?

- Application Programming Interface
- An interface for two software components to talk to each other.
- \circ What data they send
- \circ $\,$ How it's formatted $\,$
- What responses they expect.





Restaurant menu





What is HTTP ?





API Protocols





REST (REpresentational State Transfer) **API**

- Commonly used way of writing web APIs using HTTP.
- Uses the concept of resources
- Resources can be accessed via verbs and resource paths



Components of a REST API Request

• Endpoint (base path and the resource path) The URL where the request is sent (e.g., https://api.example.com/users/123).

• HTTP Methods

Defines the type of action to perform

Method	Purpose	Example
GET	Retrieve data from the server	Fetching user information
POST	Submit new data to the server	Creating a new user
DELETE	Remove data from the server	Deleting a user
PUT	Update existing data on the server	Updating user details

Components of a REST API Request

• Query Parameters (Optional) A part of the URL used to filter, sort or search data https://api.example.com/users/123?role=admin

• Headers

Additional information about the request E.g:

Content-Type - Specifies the data format (e.g., **application/json**) *Authorization -* Sends a token for authentication

• **Body/Payload** (Optional)

Contains the data to send with the request. Used in methods like POST and PUT.





Components of a REST API Response

- Status Code
 - Indicates the result of the request

E.g: 200 OK: The request was successful. 404 Not Found: The resource was not found. 500 Internal Server Error: Something went wrong on the server side. 400 Bad Request: The request was invalid (e.g., missing parameters).

• Headers

Additional information about the response E.g: *Content-Type -* Specifies the data format (e.g., **application/json**)

• Body

The data returned by the server



GET request examples

Scenario: Fetching details about a specific user

Request: Response: Status Code: 200 OK curl -X GET https://api.example.com/users/123 Content-Type: curl -X GET Body: https://api.example.com/users/123?role=admin "id": "124". "name": "Bob", "email": "bob@gmail.com"

POST request example

Scenario: Adding a new user		
Request:	Response:	
curl -X POST https://api.example.com/users \ -H "Content-Type: application/json" \	Status Code: 201 Created Content-Type: application/json*	
-d '{ "name": "Bob", "email": "bob@gmail.com"}'	{ "id": "124", "name": "Bob", "email": "bob@gmail.com" }	

PUT request example

Scenario: Update existing data on the server

Request:

- curl -X PUT https://api.example.com/users/123 \
- -H "Content-Type: application/json" \

-d '{"role": "editor"}'

Response:

Status Code: 200 OK

Content-Type: application/json

Body:

"id": "124", "name": "Bob", "email": "bob@gmail.com", "role" : "editor"



DELETE request example

Scenario: Deleting a user		
Request:	Response:	
curl -X DELETE https://api.example.com/users/123	Status Code: 200 OK Body: { "message": "User deleted successfully" }	

Integration



"Integration like putting together a jigsaw puzzle.

It's when we make different parts fit together, so the whole thing works nicely!"



Ballerina Swan Lake

- → Fully open-source programming language, powered by WSO2
- \rightarrow 6+ years of effort with 300+ contributors
- Cloud-native programming language optimized for integration
- → Both textual syntax and graphical form
- → Built-in data types suitable for network communication



Understanding Ballerina Basics



Understanding Ballerina Basics: Hello World!

- Execute the **\$ bal new hello-world** to create a new Ballerina package
- Code:

```
import ballerina/io;
public function main() {
    io:println("Hello, World!");
}
```

- The main function is the entry point of a Ballerina program
- Execute \$ bal run to run the program



Understanding Ballerina Basics: Data Types

- **JSON**: Used to send data over the network. Union of simple basic types
- ()|boolean|int|float|decimal|string|json[]|m ap<json>
- XML: A markup language and file format for storing, transmitting, and reconstructing arbitrary data

```
json profile = {
    name: "John Doe",
    age: 30,
    address: {
        city: "Colombo",
        country: "Sri Lanka"
    }
};
xml x1 = xml `<book>The Lost World</book>`;
```

Understanding Ballerina Basics: Data

- **Record**: A collection of specific named fields where each field has a type for its value.
- **Query Expressions:** A query expression is similar to the SQL query syntax where you can construct a list, a mapping, a table, a stream, or a sequence by iterating over an iterable value

```
type Address record {
    int number;
    string street;
    string city;
};
string[] upperNames = from var user in users
    where user.age > 28
    select user.name.toUpperAscii();
```



Networking in Ballerina: Services

- The service and listener are built-in constructs in Ballerina
- They provide an easy way to write network endpoints that serves client requests.

```
import ballerina/http;
service on new http:Listener(9090) {
    resource function get greeting() returns string {
        return "Hello, World!";
    }
}
```



Networking in Ballerina: Clients

- The client is also a built-in construct in Ballerina
- Clients provide an easy way to consume services

```
import ballerina/http;
import ballerina/io;
public function main() returns error? {
    http:Client greetingClient = check new("http://localhost:9090")
    String greeting = check greetingClient->/greeting;
    io:println(greeting);
}
```



Swagger/OpenAPI Specification

• A framework for describing REST APIs that supports auto-generating documentation and code.

Why Use It?

- Provides an interactive interface for testing APIs.
- Allows service consumers to understand how to interact with it.

GraphQL (Graph Query Language)

- Relatively new protocol developed by META.
- Query language for APIs.
- Has a single endpoint and always use HTTP POST method.





Nothing More Nothing Less



Nothing Less

Imagine an application that displays user profiles along with a list of their recent blog posts.

REST Approach:

- 1. **Fetch User Details:** Make a `GET` request to `/users/{userId}` to get basic user information (name, email, etc.).
- 2. Fetch User's Posts: Make another `GET` request to `/users/{userId}/posts` to get the user's blog posts.

This involves two separate network requests.



When to use REST vs GraphQL

Consider GraphQL

- Complex, Nested Data Fetching Needs
- Slow internet connection
- Your Client requirements evolve rapidly



Monolithic vs. Microservice Architecture





Monolithic Architecture

Microservice Architecture

Introduction to Microservices Architecture

- Characteristics of Microservices
 - Autonomous
 - Specialized

- Benefits of Microservices
 - Agility
 - Flexibility of scaling
 - Easy deployment
 - Technological freedom
 - Resilience



Architecture





Real-World Examples







Generative AI Applications





Bakerina Swan Lake

Vector Search



Ba[#]erina _{SwanLake}

https://medium.com/@kbdhunga/a-beginners-guide-to-similarity-search-vector-indexing-part-one-9cf5e9171976

Agents

- When you want the AI to reason and perform actions continuously.
- When you want to let the AI determine the execution order.
- When you want to maintain internal State

Building Blocks

- Model The reasoning Engine
 - System Prompt
- Tools The capabilities we give
 - Tool Name
 - Parameter Names and descriptions



How it Works



https://platform.openai.com/docs/guides/function-calling?api-mode=responses



Hands on Session - Agents

Source Code - https://github.com/xlight05/apiit_session



Questions?



Your Feedback Matters

https://forms.gle/TyVnqkQedJmqx31dA



Find out more...

- Learn Ballerina:
 - Ballerina By Example
 - <u>https://ballerina.io/learn/by-example/</u>
 - API Documentation
 - https://lib.ballerina.io/

• Join the Ballerina community







ballerina-lang

<u>ballerinalang</u>

<u>ballerina</u>

@ballerinalang



Thank you!

If you have any further questions, please raise them in the **Ballerina Discord server**.

