Ballerina

Swan Lake

Mastering API integrations and Microservices Architecture

Hello!

Niveathika Rajendran

niveathika@wso2.com | Associate Technical Lead **|** **@ballerinalang** **|** **WSO2**

Sasindu Alahakoon

sasindu@wso2.com | Senior Software Engineer **|** **@ballerinalang** **|** **WSO2**

Shammi Kolonne

shammik@wso2.com | Senior Software Engineer **|** **@ballerinalang** **|** **WSO2**

Our mission

Help our customers

Create awesome digital experiences quickly, easily, and securely

by simplifying complex technology to the point where they can:

JUST ADD DEVELOPERS.

800+ Employees
45% Engineering

Colombo, Dubai, Mumbai, Munich, London, Santa Clara, Austin, São Paulo, Sydney

18+ Years
In the Industry

700+ Customers
Across 90 Countries and 6 Continents

Rapidly Growing
Over 30% YoY Growth in ARR

Ballerina
Swan Lake

WSO2

Code Challenge

Win a Tesla Cybertruck!

(or $100,000)

DEVELOP AND DEPLOY

Use any language, any IDE, and GitHub to develop your app and run it in Choreo for free.

FRONTEND AND BACKEND

Your app needs both a frontend as well as backend APIs.

DEPLOY TO PRODUCTION

Promote your app to the Choreo production environment.

MAXIMIZE YOUR CHANCES OF WINNING

Boost your odds by enhancing your app with the following additions:

- Use a connection to integrate the backend API to the frontend
- More components*:
  - Databases
  - Manual or scheduled jobs
  - Multiple projects
  - Internal and external APIs
- Use Asgardeo for app authentication
- Use Ballerina to implement backend logic or APIs

The more you do, the greater your chances of winning the Cybertruck or $100,000!

Each Addition = 1 Additional Entry

** Up to the free tier limit*

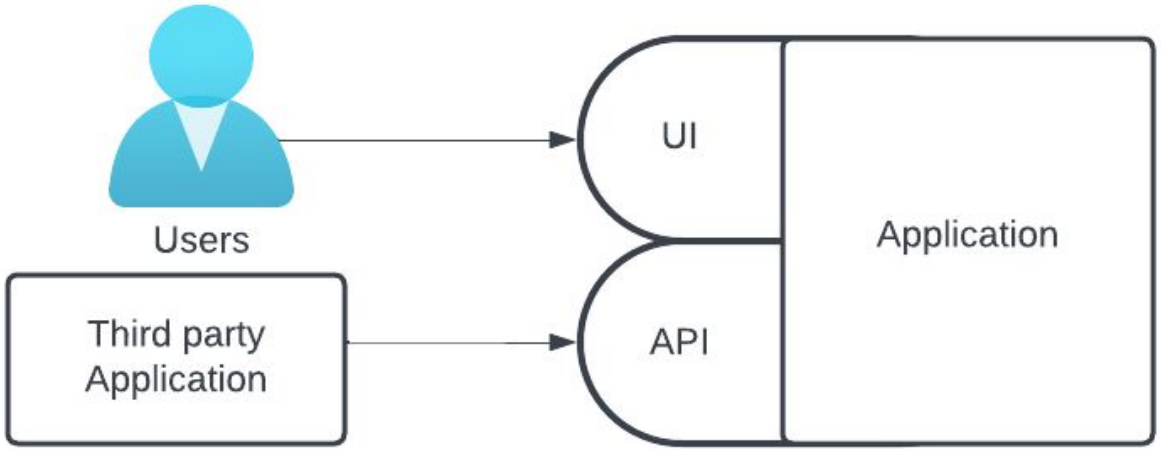
Learn more:
https://choreo.dev/cybertruck

About this Session

Coming Up

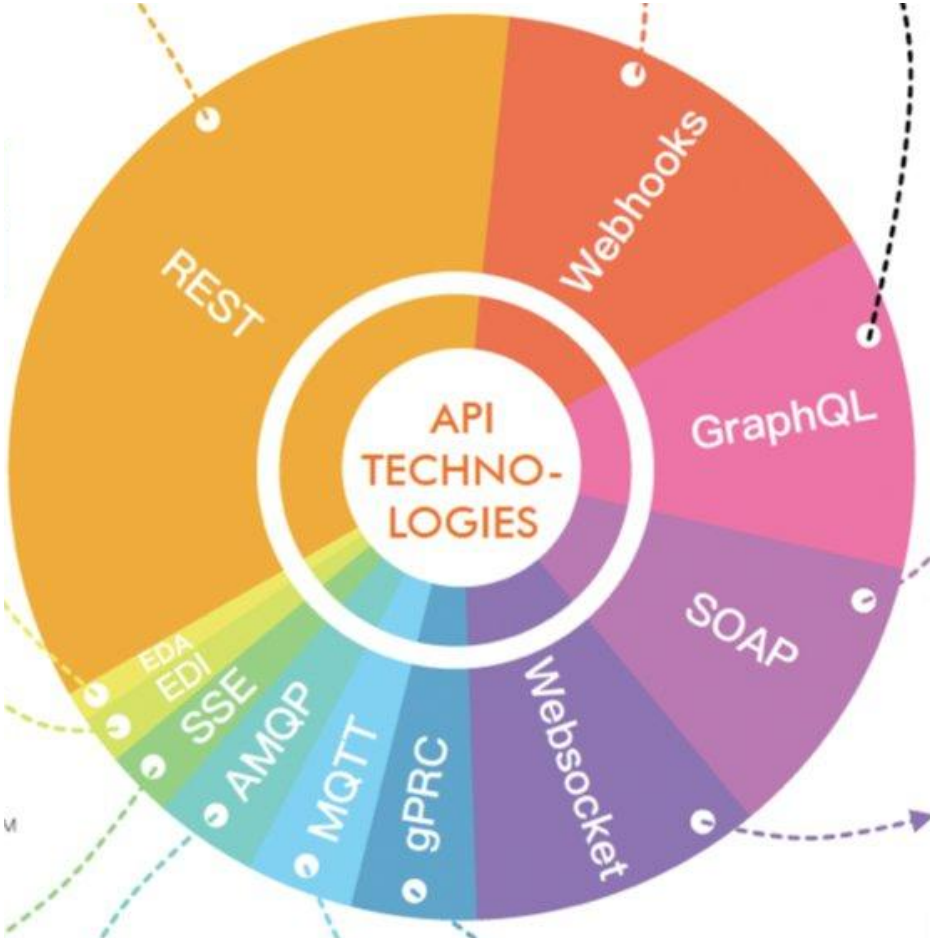
1. API Fundamentals
2. Introduction to Microservice Architecture
3. The perfect fit for effortless Integrations: Ballerina coming into the picture
4. Mastering fundamental concepts of Ballerina
5. Hands-on Session
6. The Rewards Challenge

What is an API?



What is HTTP ?

API Protocols



Ballerina
Swan Lake

API Fundamentals

REST (**RE**presentational **S**tate **T**ransfer)

- Most widely used architectural style
- Uses the concept of resources
- Resources can be accessed via verbs and resource paths
- Each resource has a standard format to represent data; server sends - client understands

API Fundamentals

GraphQL

- Relatively new protocol developed by Facebook
- Fast adaptation from the major companies
- Query language for APIs
- Data is structured as a hierarchical structure
- Has a single endpoint
- Clients can request exactly what they want, server responds with exactly what was requested

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

Real-World Examples

amazon

Uber

Integration

“Integration like putting together a jigsaw puzzle.

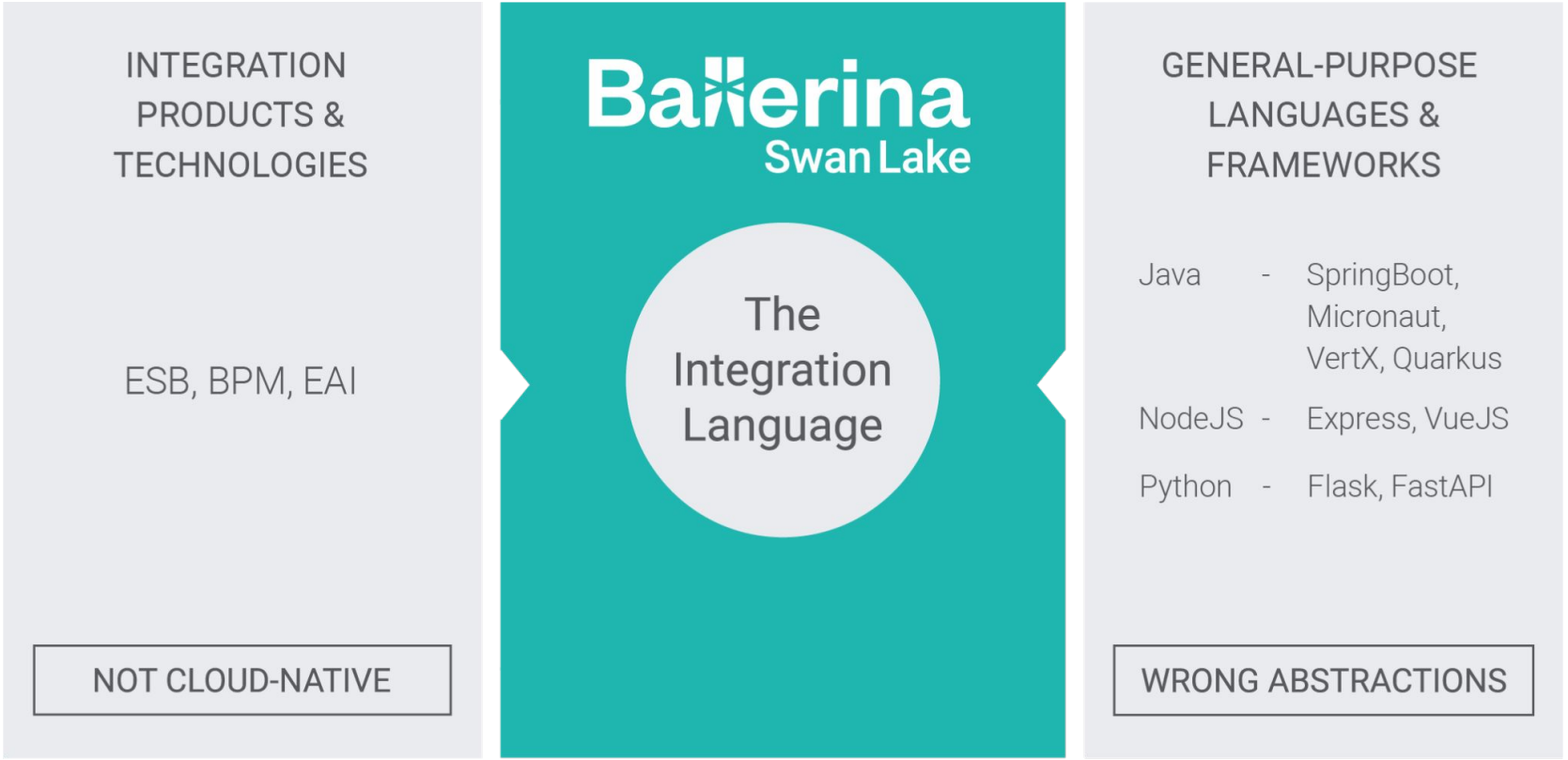
It's when we make different parts fit together, so the whole thing works nicely!”

Ballerina
Swan Lake

Ballerina Swan Lake

- ➔ Fully open-source programming language, powered by WSO2
- ➔ 6+ years of effort with 300+ contributors
- ➔ Cloud-native programming language optimized for integration
- ➔ Both textual syntax and graphical form
- ➔ Data Oriented Programming (DOP) paradigm

Addressing the integration gap

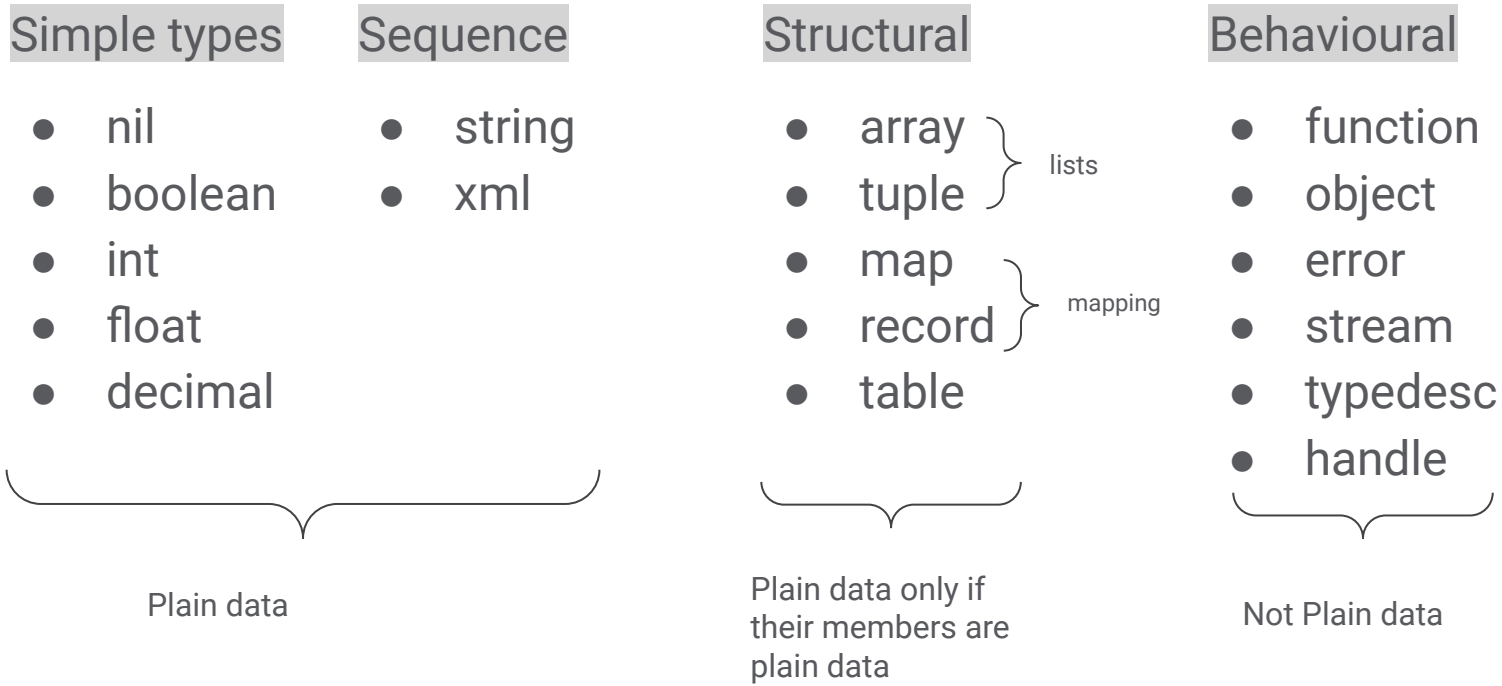


Ballerina for Integration

- Language made specifically for integration and microservices
- First class support for network endpoints
- Rich library - A collection of packages to help writing and connecting to various endpoints
- Built-in data types suitable for network communication

Understanding Ballerina Basics

Ballerina Basic Types



anydata - Type of Plain data
any - any value except for error

Understanding Ballerina Basics: Data Types

- **int**: Integer data type (64-bit signed integer)
- **float**: Floating-point data type (64-bit double-precision floating-point)
- **boolean**: Boolean data type (true or false)
- **string**: String data type (a sequence of Unicode characters)
- **Arrays**: An array can be used to hold a list of values of a given type
- **Maps**: The `map<T>` type is a data structure to store key-value pairs, with a `string` key and a value of a given type

```
// Integer
int i = 10;

// Float
float f = 12.34;

// Boolean
boolean b = true;

// String
string s = "Hello World!";

// Array of Strings
string[] names = ["John", "Doe", "Jane", "Doe"];


// Map of integers
map<int> ages = {
    "John": 30,
    "Jane": 20,
    "Karen": 40
};
```

Understanding Ballerina Basics: Data Types

- **nil**: Ballerina's version of `null` is called nil and written as `()`
- **Union Types**: `T1|T2` is the union of the sets described by `T1` and `T2`
- **Optional Types**: `T?` means the union of `T` and `()` equivalent to `T|()`
- **any**: Union type containing all the Ballerina types

```
// Nil
var n = ();

// Union (either string or int)
string|int x = 10;

// Optional (either string or nil)
string? y = 10;

// any array
any[] data = [1, "hello", 3.4, true];
```

Understanding Ballerina Basics: Data Types

- **JSON**: Used to send data over the network. Union of simple basic types
- ()|boolean|int|float|decimal|string|json[]|map<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: Records and Objects

- **Record**: A collection of specific named fields where each field has a type for its value.
- **Object**: Type definition without any implementation. It is similar to a Java interface.

```
type Address record {
    int number;
    string street;
    string city;
};

type Animal object {
    string name;

    function run() returns int;
};
```

Understanding Ballerina Basics: Functions

- Functions are building blocks of an application
- The function keyword is used to define functions in Ballerina
- A function can have zero or more input arguments and can return a value (Not returning anything means returning nil)

```
function add(int a, int b) returns int {
    return a + b;
}
```

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

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
- Execute the $ bal new hello-world-service command to create a new Ballerina package

```
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);
}
```

Hands on Session

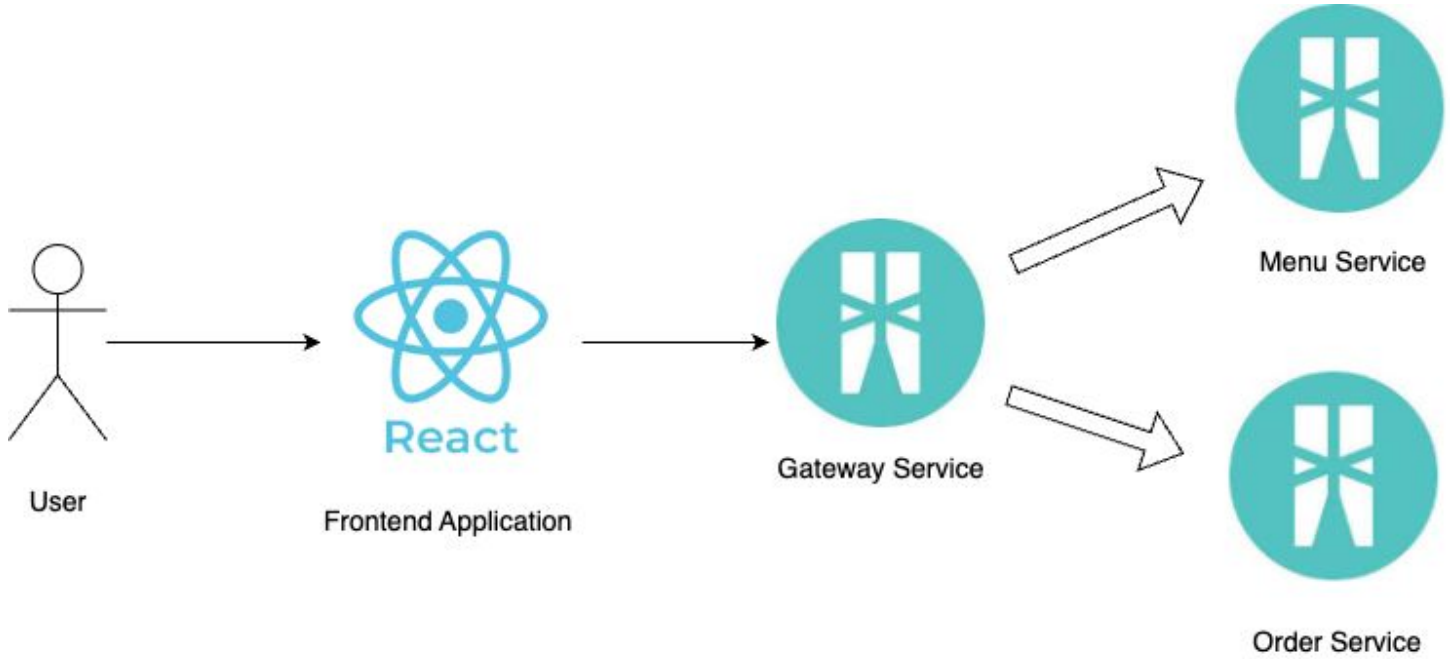
Source Code - https://github.com/SasinduDilshara/hotel_order_service

Hotel Order Service

- Based on Microservice Architecture
  - Menu Service 
  - Order Service 

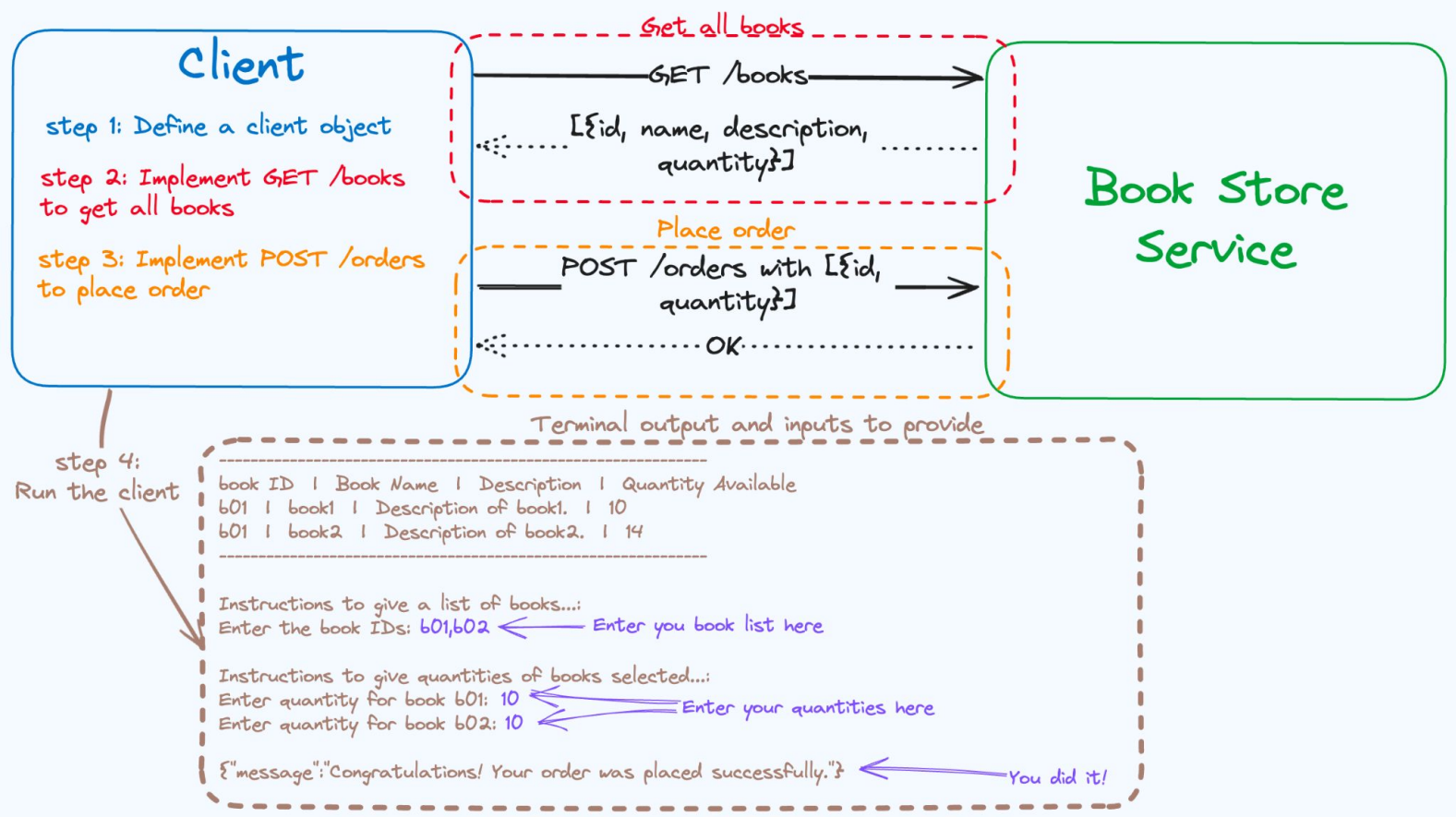
- Use cases
  - Create new orders
  - View all orders
  - View all food in the menu

Hands-on Session - Hotel Management Service



Rewards Challenge

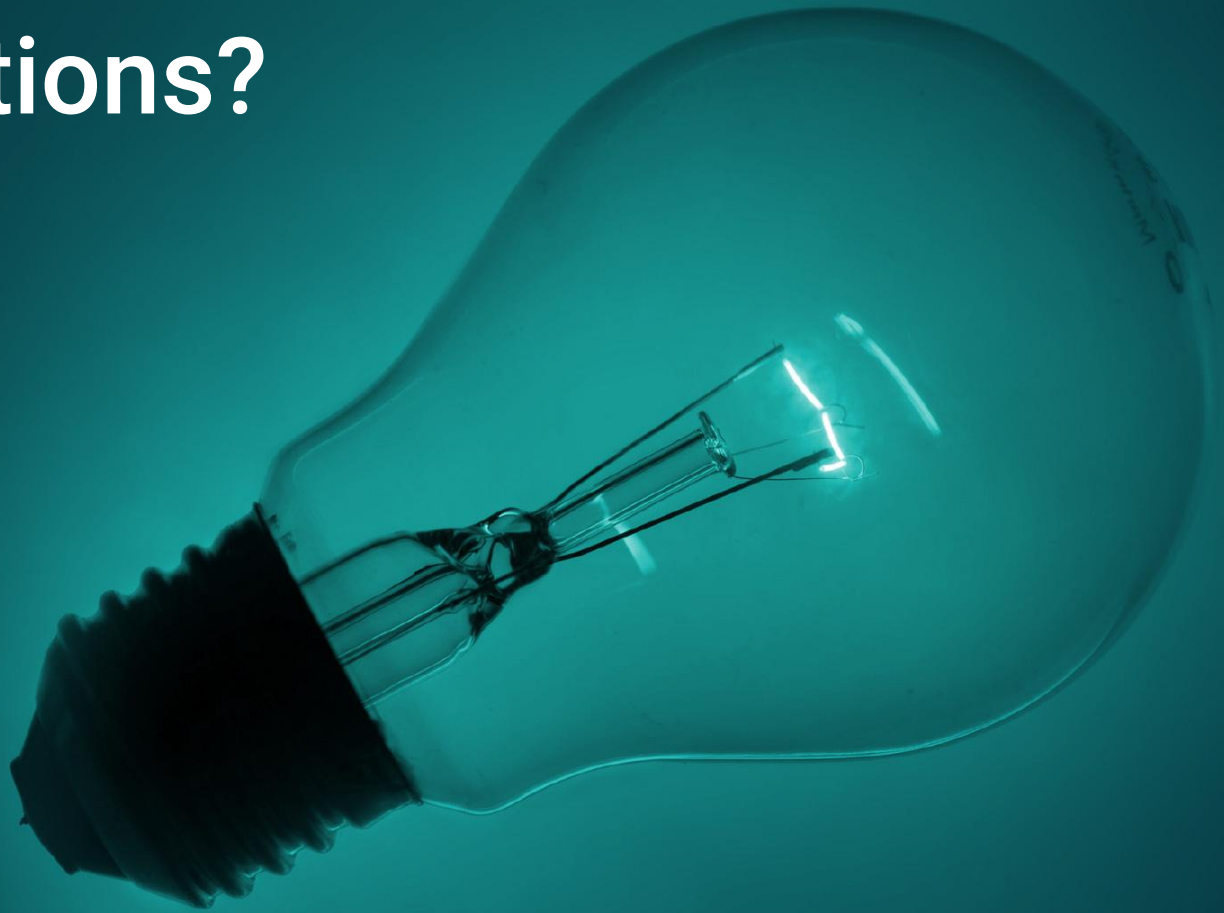
Steps to Complete



Rewards Challenge

- Implement a client to interact with the bookstore service and consume APIs to view all available books and place an order for a set of selected books.
- The partially completed client exercise, together with the bookstore service is available at https://github.com/ShammiL/bookstore.
- Access the code base using one of the following methods,
    - Fork the above repository and clone it to your machine or
    - Download as a ZIP
- Students who completed the task will get a **special reward!**
- The completed code should be pushed to a GitHub repository and the link should be shared with us to be eligible for the rewards.
- Winners will be decided by a panel of judges. The judges' decision will be final.

Questions?



Your Feedback Matters

https://forms.gle/PW4tPiRz48X24j798

Mini Project

- Do something cool with/about Ballerina
    - A new Ballerina package, published to Ballerina central
    - An article/video about Ballerina
    - Contribute to Ballerina project (Find “Good First Issues”)
    - Make sure your source code/article/video is public
- **Submit your projects** using the below google form
    - https://forms.gle/nopCp3utp7FG3Loq8
- There’s no limit, submit as many entries as you want
- Successful submissions will receive free vouchers for WSO2 practitioner and developer certifications.

Find out more...

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

- Join the Ballerina community

ballerinalang

ballerina

@ballerinalang

ballerina-lang

Thank you!

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