Ballerina

Swan Lake

Backend Development with Ballerina

Hello!

Gayal Dassanayake

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

Shammi Kolonne

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

Understanding Ballerina Basics

Ballerina
Swan Lake

Ballerina Basic Types

Simple types

- nil
- boolean
- int
- float
- decimal

Sequence

- string
- xml

Plain data

Structural

- array
- tuple

lists

- map
- record

mapping

- table

Plain data only if their members are plain data

Behavioural

- function
- object
- error
- stream
- typedesc
- handle

Not Plain data

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
};
```

Ballerina
Swan Lake

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];
```

Ballerina Swan Lake

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>`;
```

Ballerina Swan Lake

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;
};
```

Ballerina
Swan Lake

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;
}
```

Ballerina
Swan Lake

Understanding Ballerina Basics: Hello World!

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

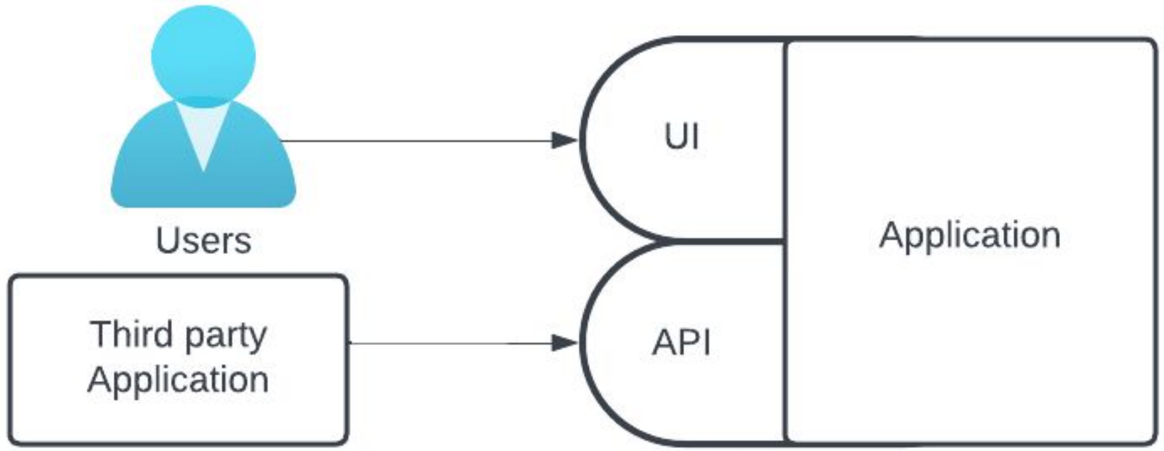
```
import ballerina/io;


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 REST Services

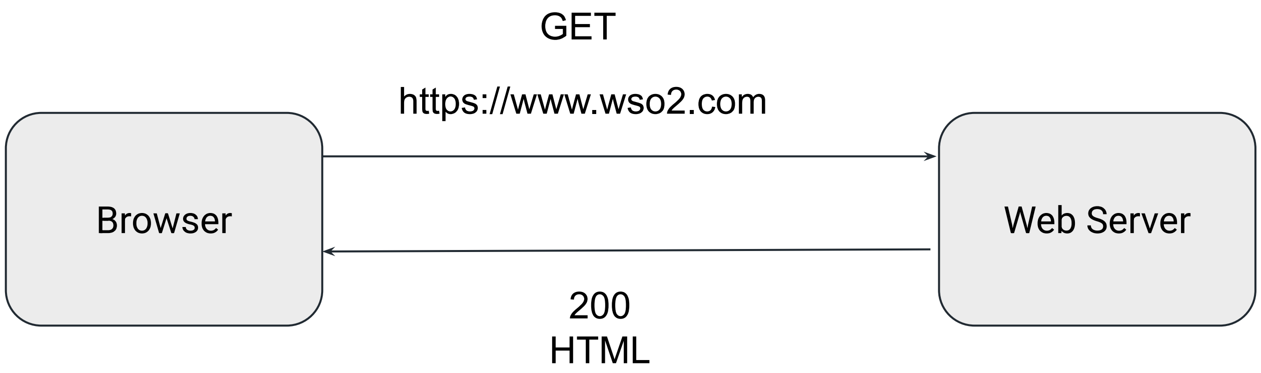
Ballerina
Swan Lake

What is an API?



Ballerina Swan Lake

What is HTTP ?



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

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!";
    }
}
```

Ballerina
Swan Lake

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);
}
```

Ballerina
Swan Lake

Understanding Ballerina Services: Hello World!

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

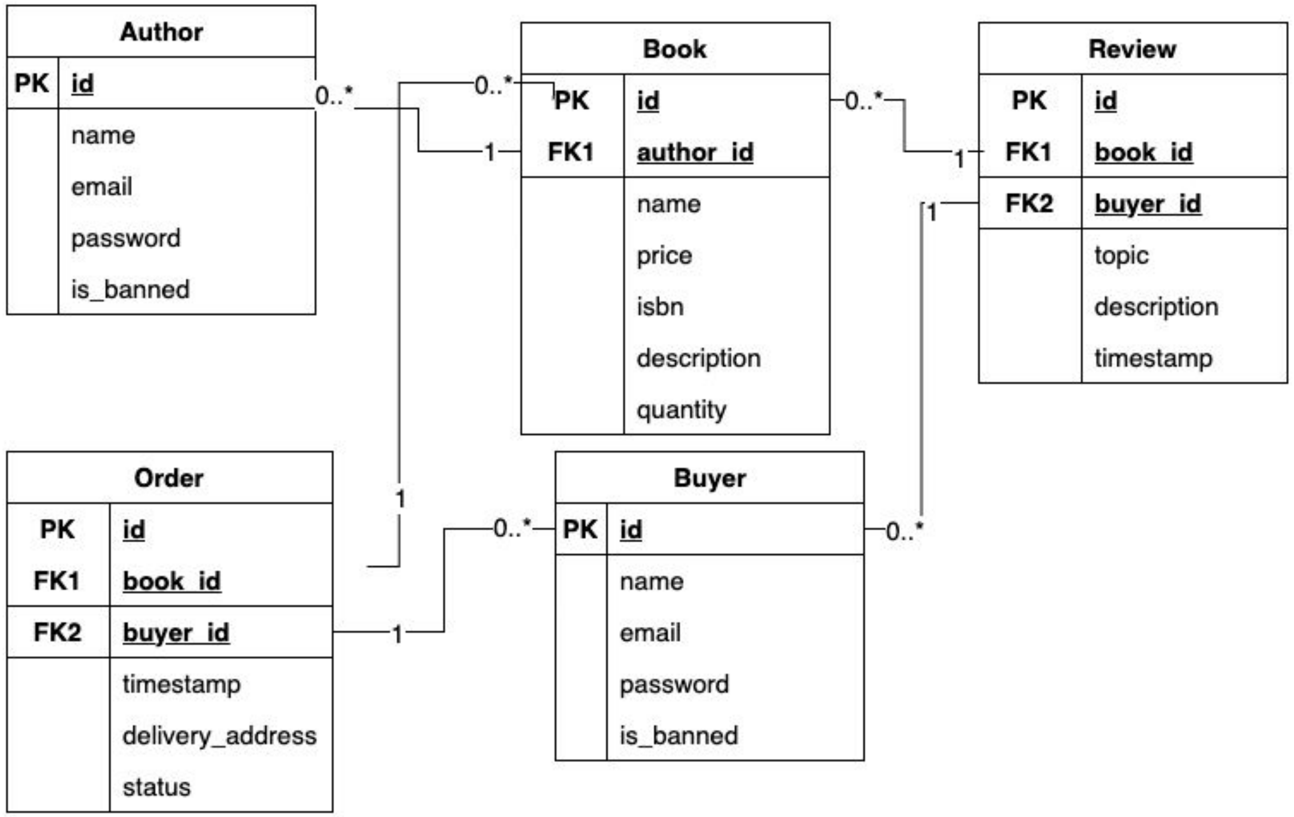
```
import ballerina/http;

service / on new http:Listener(9090) {
   resource function get greeting(string? name) returns string|error {
       if name is () {
           return error("name should not be empty!");
       }
       return string `Hello, ${name}`;
   }
}
```

- Execute `$ bal run` to run the program
- Send a GET request to the service through curl:
  - Curl : `curl --location 'http://localhost:9090/greeting?name=gayal'`

Ballerina Swan Lake

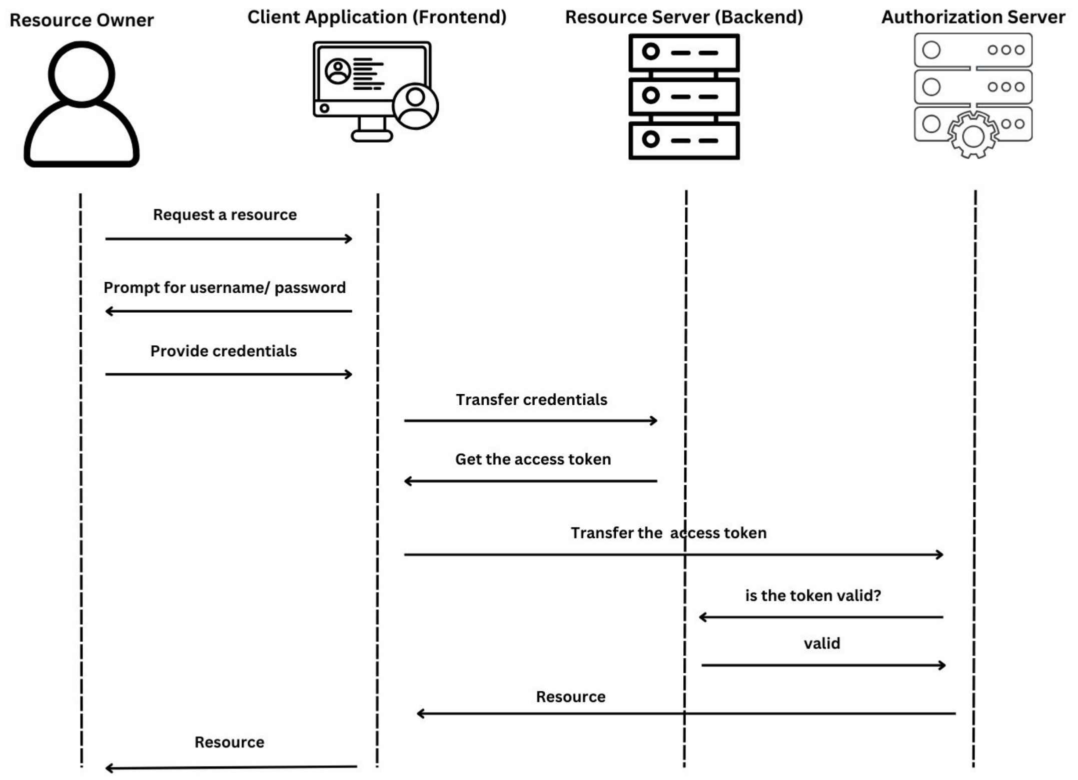
Hands-on Session

Overview of the Book Marketplace System



Ballerina Swan Lake

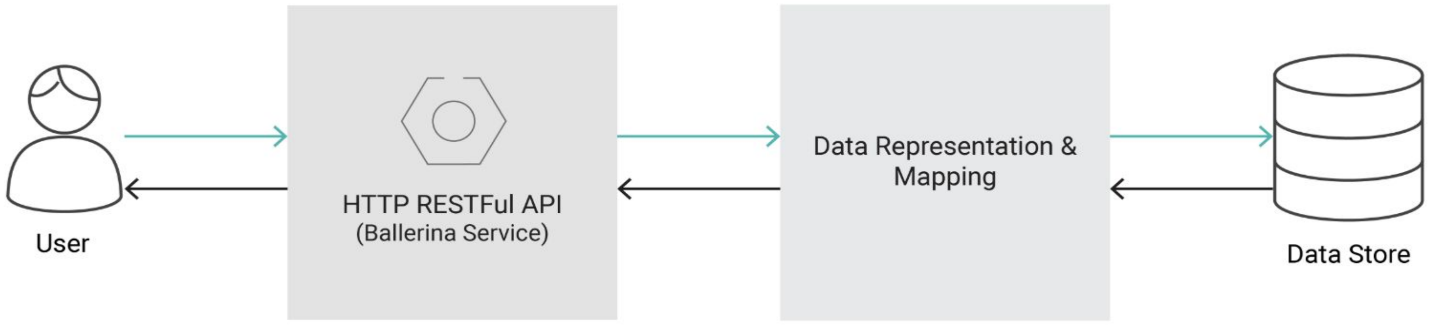
OAuth


Resource Owner
Client Application (Frontend)
Resource Server (Backend)
Authorization Server
Request a resource
Prompt for username/ password
Provide credentials
Transfer credentials
Get the access token
Transfer the access token
is the token valid?
valid
Resource
Resource

```
@http:ServiceConfig {
    auth: [
        {
            oauth2IntrospectionConfig: {
                url: "https://localhost:9445/oauth2/introspect", // URL of the sts server
                tokenTypeHint: "access_token",
                scopeKey: "scp",
                clientConfig: {
                    customHeaders: {"Authorization": "Basic YWRtaW46YWRtaW4="},
                    secureSocket: {
                        cert: "/path/to/public.crt"
                    }
                }
            },
            scopes: "admin"
        }
    ]
}
```

Ballerina
Swan Lake

Ballerina Persist



- Manage data persistence easily.
- Define only a data model to generate records and client APIs instead of SQL queries.
- Generates the SQL scripts to setup the database.

Ballerina Connectors

```
// redis
redis:Client redis = check new (connection = { host: "localhost", port: 6379});
check redis->set("key", "value");

// github
github:Client github = check new (gitHubConfig);
github:Repository[] userRepos = check github->/user/repos(visibility = "private", 'type = ());

// twilio
twilio:CreateMessageRequest messageRequest = {
   To: "+XXXXXXXXXXX", From: "+XXXXXXXXXXX", Body: "Hello from Ballerina"
};
twilio:Message response = check twilio->createMessage(messageRequest);
```

Ballerina
Swan Lake

Support for more than 500+ SAAS connectors

https://github.com/gayaldassanayake/book-marketplace