Ballerina

An Introduction to Ballerina

September 2023

Who am I ?

Dilan Perera

Software Engineer at WSO2 | Developer at Ballerina

BSc (Honors) in Computer Science
University of Colombo School of Computing

What is Ballerina ?

- Open source, cloud-native programming language optimized for integration
- Rich ecosystem of
  - Data formats
  - Network protocols
  - Connectors
- Developed by WSO2 since 2016 and first released in February 2022

Ballerina fills the Integration Gap

What is Integration?

The process of connecting different software systems, applications, or components in a way that allows them to work together and exchange data called Integration.

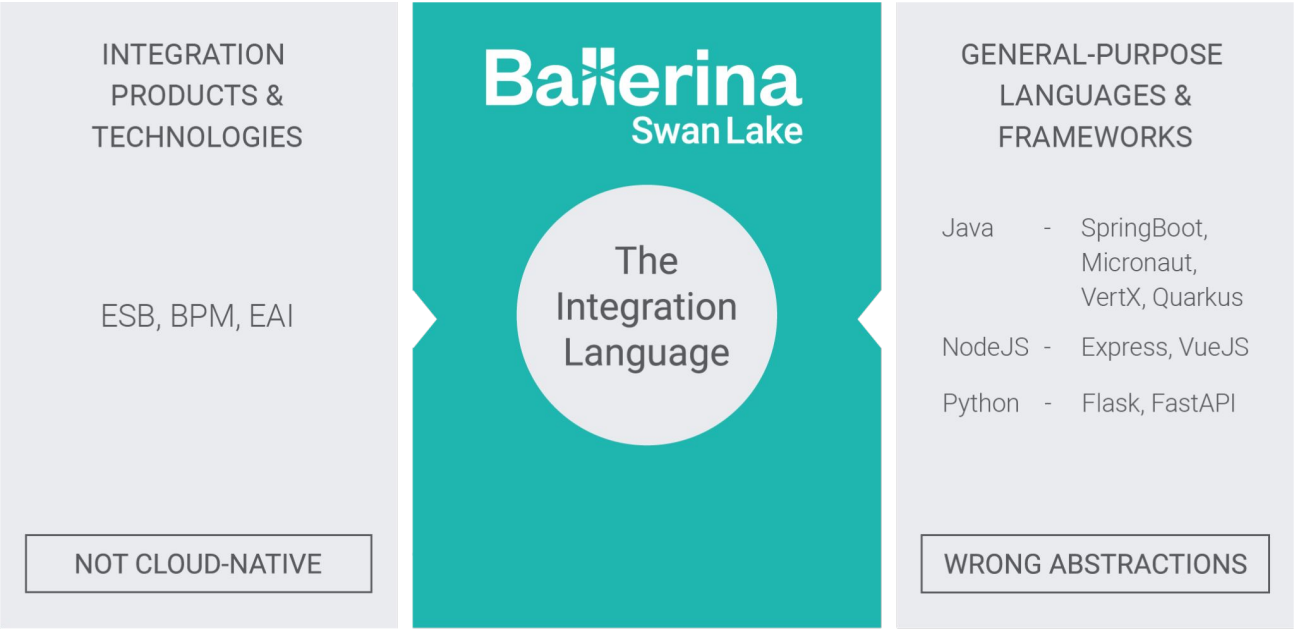
- API Integration: Integrating third-party APIs into your application to access external services or data.
  Eg - Integrating a payment gateway API(PayPal,PayHere) to your application

- Database Integration: Connecting and integrating with databases to retrieve, store, or manipulate data.
  Eg - Connecting DBs like MySQL, PostgreSQL, MongoDB, etc.., to your application.

- Middleware Integration: Integrating middleware components such as message brokers
  Eg - Integrating Azure Service Bus to build a scalable and reliable order processing system.

Ballerina

What is Integration Gap?



Let’s Explore Ballerina…

Data Types

Data Types in Ballerina

int: Integer data type (32-bit signed integer).

float: Floating-point data type (64-bit double-precision floating-point).

decimal: Decimal data type for precise decimal arithmetic.

boolean: Boolean data type (true or false).

string: String data type (a sequence of Unicode characters).

nill: 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.

```
//Integer
int i = 10;

//Float
float f = 10.0;

//Decimal
decimal d = 10.0;

//Boolean
boolean b = true;

//String
string s = "Hello World";

//Nill (value of n can be int or nill)
int? n = ();

//Union of string and int
string|int x = 10;
```

Ballerina

Data Types in Ballerina

Arrays: An array can be used to hold a set of values of the same type.

Maps: The map<T> type is a mapping from strings to T.

anydata: The top-level type representing any data type.

Other Types : table,stream,byte,error,enum,RegExp,etc..,

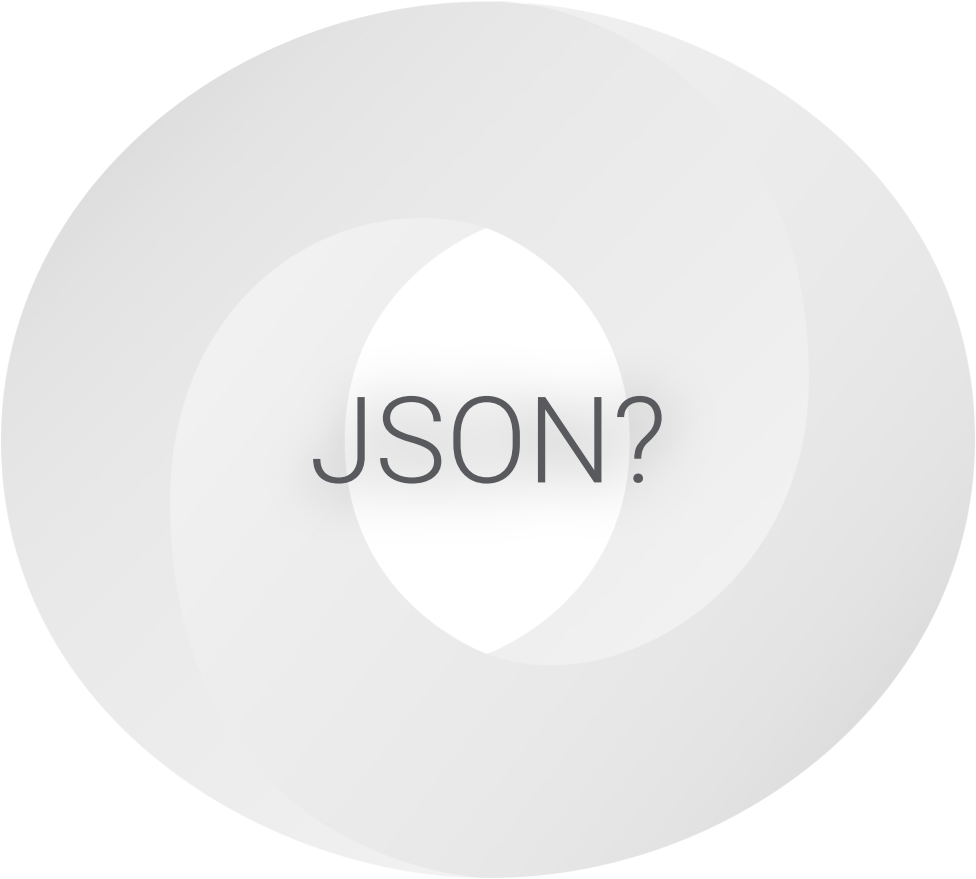
```
// Declare an int array of length 3.
int[3] numbers = [1, 2, 3];

// Declare a variable-length array.
string[] names = ["Mike", "Amy", "Korina"];

// Creates a `map` constrained by the `int` type.
map<int> ages = {
    "Tom": 23,
    "Jack": 34
};

// anydata is a union of all data types
anydata ad = [1, "Hello", 10.0, true];
```

Ballerina


JSON?

Introducing Ballerina Records

Ballerina Records

Network Data = Program Data

Processing data coming or going over the wire is a no-brainer with Ballerina. Seamlessly and selectively map network data into domain types for a range of formats, including JSON, EDI, and XML.

```
type Author record {
    string authorId;
    string firstName;
    string lastName;
};

type Book record {
    string bookId;
    string name;
    Author author;
};

type Member record {
    string memberId;
    string firstName;
    string lastName;
    Book[] books;
};
```

```
Member kelly = {
    memberId: "M001",
    firstName: "Kelly",
    lastName: "Clarkson",
    books: [
        {
            bookId: "B001",
            name: "Harry Potter and the Philosopher's Stone",
            author: {
                authorId: "A001",
                firstName: "J. K.",
                lastName: "Rowling"
            }
        },
        {
            bookId: "B002",
            name: "Harry Potter and the Chamber of Secrets",
            author: {
                authorId: "A001",
                firstName: "J. K.",
                lastName: "Rowling"
            }
        }
    ]
};
```

How Code Looks Like?

Hello from Ballerina

- The main function is an entry point to a Ballerina program
- Many syntactic and semantic similarities between Ballerina and C-family languages including Java
- Designed to maintain familiarity where possible
- Inspired by many languages

```
import ballerina/io;

public function main() {
    io:println("Hello World");
}
```

Ballerina

Consuming services

```
import ballerina/http;
import ballerina/io;

type JokeType record{
    string category;
    string joke;
};
public function main() returns error?{

    // Create an HTTP Client to make a request
    http:Client jokeClient = check new("https://v2.jokeapi.dev/joke/Any?format=txt");

    // Make the HTTP request to fetch a joke
    JokeType responce = check jokeClient->/;

    // Print the joke
    io:println(responce.joke);
}
```

Simple HTTP Service

```
import ballerina/http;

service / on new http:Listener(8080) {
    resource function get greeting() returns string {
        return "Hello World!";
    }
    resource function get greeting/[string name]() returns string {
        return "Hello " + name;
    }
}
```

```
Terminal$ curl -v http://localhost:8080/greeting
*   Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /greeting/Ballerina HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.1.2
> Accept: */*
>
< HTTP/1.1 200 OK
< content-type: text/plain
< content-length: 15
< server: ballerina
< date: Tue, 19 Sep 2023 12:16:41 +0530
<
* Connection #0 to host localhost left intact
Hello World
```

```
Terminal$ curl -v http://localhost:8080/greeting/Ballerina
*   Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /greeting/Ballerina HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.1.2
> Accept: */*
>
< HTTP/1.1 200 OK
< content-type: text/plain
< content-length: 15
< server: ballerina
< date: Tue, 19 Sep 2023 12:16:41 +0530
<
* Connection #0 to host localhost left intact
Hello Ballerina
```

Why Ballerina?

What is special about Ballerina?

Ballerina Central

Connect with anything

Ballerina

Ballerina Central

Access thousands of connectors for HTTP APIs (OpenAPI), event APIs (AsyncAPI), GraphQL services, legacy systems, and data stores, allowing seamless data transfer to and from any system, anywhere.

https://central.ballerina.io

BallerinaCentral

What is Ballerina? Swan Lake Sign in / Sign up

Discover Ballerina packages of reusable code, and assemble them in powerful ways

eg: kafka or ballerinax/kafka or org:ballerinax Search

Ballerina Central is a globally hosted package management system that is used to discover, download, and publish packages.

Popular packages

How to push a package

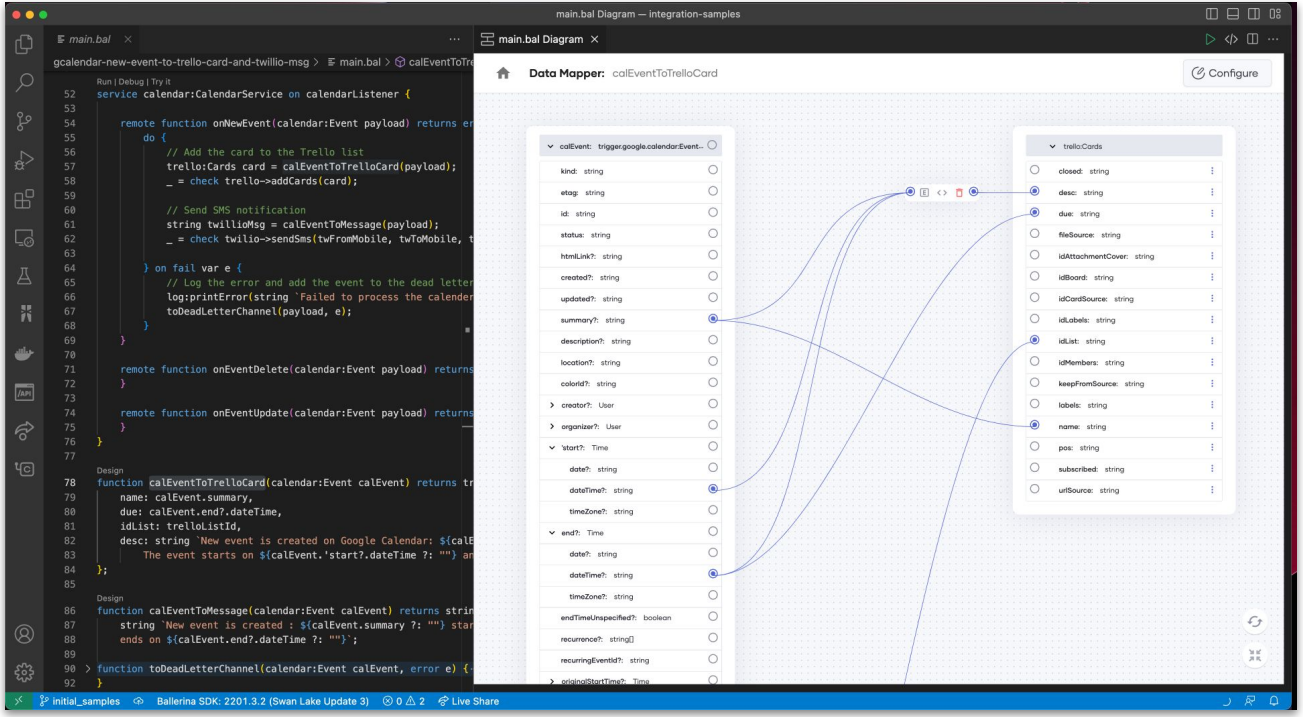
ballerinax/choreo	ballerinax/client.config	ballerinax/asyncapi.native.handler	ballerinax/googleapis.sheets
0.4.12	1.0.1	0.2.0	3.4.0
java11 317552 25.08.22	any 55402 05.10.22	java11 27947 19.12.21	java11 6148 18.05.23

Ballerina Data Mapper

Ballerina

Data transformations

Ballerina has cracked the challenge of mapping one kind of data value to another kind of data value, simultaneously as code and picture, so that both are simple, powerful, and boundless.



Ballerina

Ballerina Built-in Transactions

Built-in transactions

Eventual consistency in Data integration is nice and all.

but if you really need to make sure your distributed data integrations run transactionally

then Ballerina makes that effortless and mistake-free for developers with compile-time support.

```
type Order record {|
    string id;
    string orderDate;
    string productId;
    int quantity;
|};

final mysql:Client db = check new (host, user, password, database, port);

function createOrder(Order 'order) returns error? {
    // Start a transaction.
    transaction {
        // Insert into `sales_order` table.
        _ = check db->execute(`INSERT INTO sales_orders VALUES (${'order.id},
                ${'order.orderDate}, ${'order.productId}, ${'order.quantity})`);

        // Update product quantity as per the order.
        sql:ExecutionResult inventoryUpdate = check db->execute(`UPDATE inventory SET
                quantity = quantity - ${'order.quantity} WHERE id = ${'order.productId}`);

        // If the product is not found, rollback or else commit the transaction.
        if inventoryUpdate.affectedRowCount == 0 {
            rollback;
            return error(string `Product ${'order.productId} not found.`);
        } else {
            check commit;
        }
    } on fail error e {
        // In case of error, the transaction block is rolled back automatically.
        return error(string `Error occurred while processing the order: ${'order.id}.`, e);
    }
}
```

Ballerina

Ballerina Concurrent Programming

Concurrent programming made simple

Sequence diagrams have been used to model concurrency for decades.

Ballerina's concurrent programming model is sequence diagrams along with various concurrency control capabilities that make concurrent programming visual and accessible to all.

```
type Quote record {
    string customerName;
    string product;
    int quantity;
    decimal price;
};

function findBestQuote(QuoteRequest quoteReq) returns Quote {
    // The fork statement starts with one or more named workers,
    //  which run in parallel with each other
    fork {
        worker venderA returns Quote|error {
            http:Client venderAEP = check new (venderAURL);
            return venderAEP -> /quote.get(p = quoteReq.product, q = quoteReq.quantity);
        }

        worker venderB returns Quote|error {
            http:Client venderBEP = check new (venderBURL);
            return venderBEP -> /quote.get(p = quoteReq.product, q = quoteReq.quantity);
        }

        worker venderC returns Quote|error {
            http:Client venderCEP = check new (venderCURL);
            return venderCEP -> /quote.get(p = quoteReq.product, q = quoteReq.quantity);
        }
    }

    // Wait for all the workers to finish and collect the results.
    map<Quote|error> quotes = wait {venderA, venderB, venderC};
    return bestQuote(quotes);
}
```

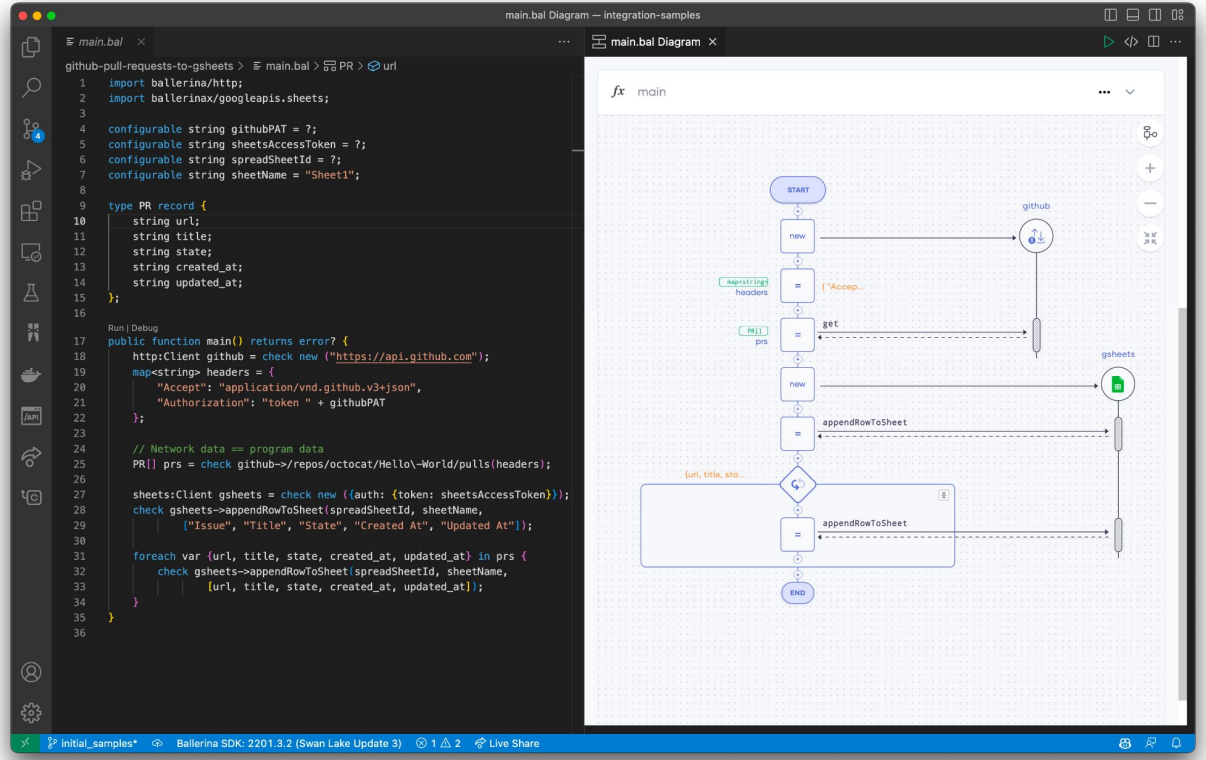
Ballerina

Ballerina Graphical

Ballerina

Code is the Picture / Picture is the Code

Instead of deciphering lines of code, Ballerina programs can be viewed and edited as sequence diagrams with flow charts. This makes maintaining and understanding integration applications a breeze. Code never goes out of sync with the picture and vice versa.

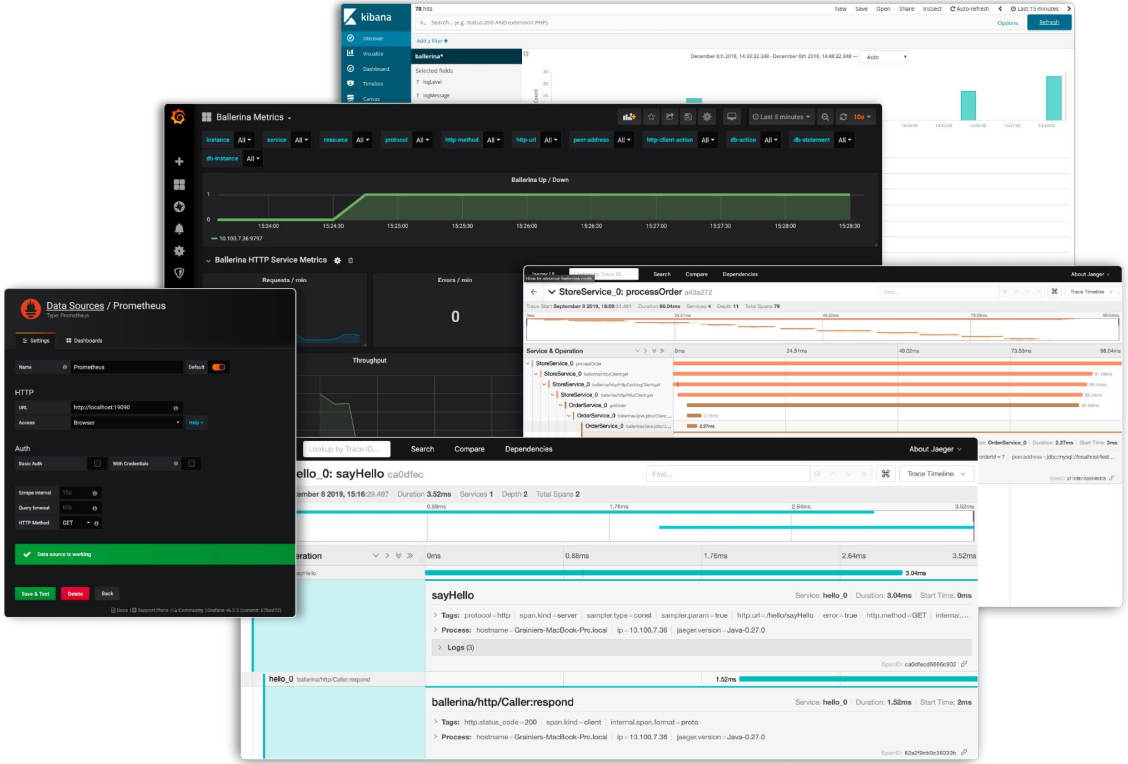


Ballerina

Ballerina Built-in Observability

Built-in observability

Every Ballerina program is automatically observable by any Open Telemetry tool, giving you complete control and visibility into the code's behavior and performance.



Ballerina

GitHub Copilot Support

Your artificially intelligent pair programmer knows Ballerina

GitHub Copilot, your artificially intelligent pair programmer

GitHub Copilot knows Ballerina. Why do all the work? Let Copilot do at least half of it.

```
3
4   type Product record {|
5       string id;
6       string name;
7       string description;
8       decimal price;
9   |};
10
11  mysql:Client productDB = new();
12
    Run | Debug | Try it
13  service / on new http:Listener(9090) {
14      resource function get products() returns Product[]|error {
            stream<Product, error?> products = productDB->query(`SELECT * FROM product`);
            return from var product in products select product;
        }

        resource function get products/[string id]() returns Product|error? {
            return productDB->queryRow(`SELECT * FROM product WHERE id = ${id}`);
        }

        resource function post products(@http:Payload Product product) returns error? {
            _ = check productDB->execute(`INSERT INTO product (id, name, description, price) VALUES (${product.id}, ${product.name}, 
        }

        resource function put products/[string id](@http:Payload Product product) returns error? {
            _ = check productDB->execute(`UPDATE product SET name = ${product.name}, description = ${product.description}, price = ${
        }

        resource function delete products/[string id]() returns error? {
            _ = check productDB->execute(`DELETE FROM product WHERE id = ${id}`);
        }
15  }
16
17
```

Ballerina

Ballerina for AI as a Service

Ballerina

Ballerina for AI as a Service

For many years Python, a wonderful language, has been the best choice for data analytics, data science, and machine learning.

Now AI available as a Service from OpenAI, Microsoft, Google, Facebook, and others.

With these LLMs,Add AI to business applications is not about those problems anymore but more about **prompt engineering, fine-tuning and calling APIs offered by hosted LLMs.**

Ballerina is your best choice for writing modern cloud-native applications that incorporate LLM-powered AI!

Azure Open AI - Ballerina Connectors

ballerinax/azure.openai.text 1.0.3

Connects to Azure OpenAI Completions API from Ballerina.

by ballerinax 18.05.23 API Documentation

ballerinax/azure.openai.chat 1.0.2

Connects to Azure OpenAI Chat Completions API from Ballerina.

by ballerinax 11.05.23 API Documentation

ballerinax/azure.openai.deployment 1.0.1

Connects to Azure OpenAI Deployments API from Ballerina.

by ballerinax 11.05.23 API Documentation

ballerinax/azure.openai.embeddings 1.0.2

Connects to Azure OpenAI Embeddings API from Ballerina.

by ballerinax 11.05.23 API Documentation

ballerinax/azure.openai.finetunes 1.0.1

Connects to Azure OpenAI Files API,

by ballerinax 11.05.23 API Documentation

Open AI - Ballerina Connectors

ballerinax/openai.chat 1.1.2

Connects to the OpenAI Chat API from Ballerina with the `ballerinax/openai.chat` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.text 1.0.5

Connects to the OpenAI Completions API from Ballerina with the `ballerinax/openai.text` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.embeddings 1.0.5

Connects to the OpenAI Embeddings API from Ballerina with the `ballerinax/openai.embeddings` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.images 1.0.5

Connects to the OpenAI Images API from Ballerina with `ballerinax/openai.images` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.audio 1.0.5

Connects to the OpenAI Audio API from Ballerina with the `ballerinax/openai.audio` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.finetunes 1.0.5

Connects to the OpenAI Fine-tunes API from Ballerina with the `ballerinax/openai.finetunes` package.

by ballerinax 17.08.23 API Documentation

ballerinax/openai.moderations 1.0.5

Connects to the OpenAI Moderations API from Ballerina with the `ballerinax/openai.moderations` package.

by ballerinax 17.08.23 API Documentation

Instant Hosting with WSO2 Choreo

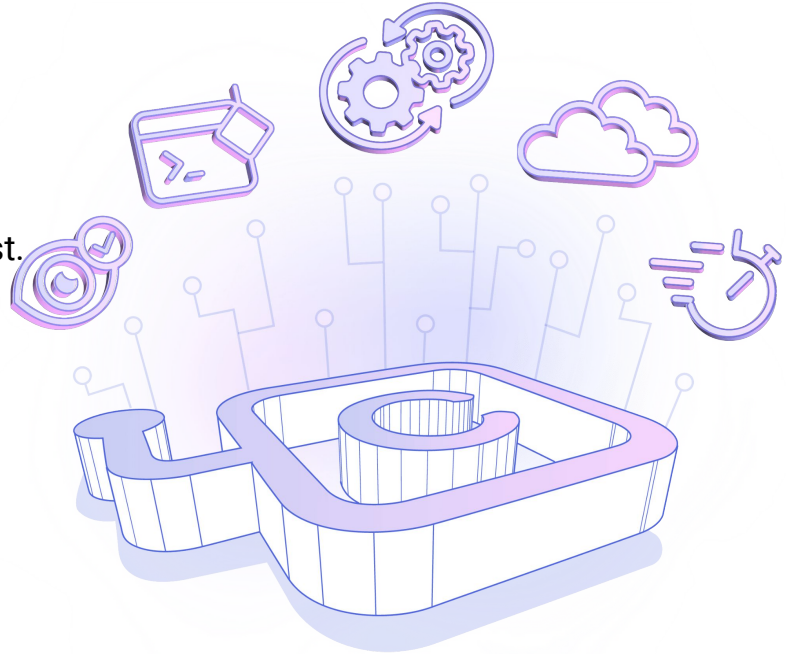
Instant Hosting with WSO2 Choreo iPaaS

Your ballerina code has,

Manual integrations?
Scheduled integrations (cron jobs)?
Triggered integrations?
Integrations as APIs?

No problem!

Write the code, attach the repo to WSO2 Choreo, and let it do the rest.



That’s it about Ballerina

Challenge to Win Gifts

Create something fascinating with Ballerina

Please send us a short video, pictures, or a document showcasing your creation.

Ensure that the source code is available in a public GitHub repository and provide the repository link along with your submission.

Creative projects will be rewarded with gifts and vouchers.

Feel free to submit as many entries as you like | Anyone can participate

Submission Link : https://bit.ly/ucscbal

Ballerina

Thank You !

For Slides,

Bakerina