WIKIPEDIA

Ballerina (programming language)

Ballerina is an open source general-purpose programming language and platform designed by WSO2 for cloud-era application programmers. It is easy to write and modify and is suitable for application programmers.[5][6][7]

It is an open source project [2] started in 2015 by architects from WSO2 as a code-based alternative to the configuration-based integration tools such as EAI, ESB, and workflow products.[8][9]

It has various constructs geared toward cloud-native development including support for modern data formats and protocols, reliability, distributed transactions, APIs, and event streams.[10][11][12]

Ballerina

Designed by	Sanjiva Weerawarana, James Clark, Sameera Jayasoma, Hasitha Aravinda, Srinath Perera, Frank Leymann and WSO2[1]
Developer	WSO2
First appeared	2017
Typing discipline	Structural, strong, static, inferred
Implementation language	Java, Ballerina, TypeScript [2]
OS	Cross-platform
License	Apache License 2.0[3]
Website	ballerina.io (http://ballerina.io/)
Influenced by	
Java, Javascript, Go, Rust, C#[4]	

Contents

History

Ballerina was designed by WSO2 to improve productivity for application developers that have to work with distributed cloud-native systems. The designers, who provided enterprise products in the integration space for over 10 years, used their knowledge of the industry when designing the language.[13][14] Ballerina was first publicly announced in 2017 and version 1.0 was released on September 10, 2019.[15]

Design

Some key concepts in Ballerina include:

- The network in the language - Ballerina introduces fundamental, new abstractions of client objects, services, resource functions, and listeners to bring networking into the language.[16]
- Sequence diagrams for programming - In Ballerina, every program has a corresponding sequence diagram that illustrates distributed and concurrent interactions automatically.[17]

- Structural, open-by-default typing - Ballerina has a statically-typed, structural type system that is designed to be network data schema friendly.[18]
- Moving from code to cloud - Ballerina brings the entire program execution process to the hands of the developer with extensible metadata that gets compiled to runnable programs for all major cloud platforms.[19]
- Automated observability - Ballerina incorporates automatic observability features into the language itself that helps keep track of metrics, logs and tracing.[20]

Examples

Hello World Service

```
import ballerina/http;

service hello on new http:Listener(9090) {

    resource function sayHello(http:Caller caller,
        http:Request req) returns error? {

        check caller->respond("Hello, World!");
    }
}
```

To start the service, navigate to the directory that contains the `.bal` file, and execute the `ballerina run` command below.

```
$ ballerina run hello_world.bal
[ballerina/http] started HTTP/WS listener 0.0.0.0:9090

curl http://localhost:9090/hello/sayHello
Hello, World!
```

[21]

Workers

```
import ballerina/http;
import ballerina/lang.'int;
import ballerina/io;

// Workers interact with each other by sending and receiving messages.
// Ballerina validates every worker interaction (send and receive)
// to avoid deadlocks.
public function main() {
    worker w1 {
        int w1val = checkpanic calculate("2*3");
        // Sends a message asynchronously to the worker `w2`.
        w1val -> w2;
        // Receives a message from the worker `w2`.
        int w2val = <- w2;
        io:println("[w1] Message from w2: ", w2val);
        // Sends messages synchronously to the worker `w3`. The worker `w1` will wait
        // until the worker `w3` receives the message.
        w1val ->> w3;
        w2val -> w3;
        // Flushes all messages sent asynchronously to the worker `w3`. The worker
        // will halt at this point until all messages are sent or until the worker `w3`
        // fails.
        checkpanic flush w3;
    }

    // A worker can have an explicit return type, or else, if a return type is not mentioned,
    // it is equivalent to returning ().
    worker w2 {
        int w2val = checkpanic calculate("17*5");
        // Receives a message from the worker `w1`.
        int w1val = <- w1;
        io:println("[w2] Message from w1: ", w1val);
        // Sends a message asynchronously to the worker `w1`.
        w1val + w2val -> w1;
    }

    worker w3 {
        int
```

[22]

gRPC Unary Blocking

```
import ballerina/grpc;
import ballerina/log;

service HelloWorld on new grpc:Listener(9090) {

    resource function hello(grpc:Caller caller, string name,
                            grpc:Headers headers) {
        log:printInfo("Server received hello from " + name);
        string message = "Hello " + name;

        // Reads custom headers in request message.
        string reqHeader = headers.get("client_header_key") ?: "none";
        log:printInfo("Server received header value: " + reqHeader);

        // Writes custom headers to response message.
        grpc:Headers resHeader = new;
        resHeader.setEntry("server_header_key", "Response Header value");

        // Sends response message with headers.
        grpc:Error? err = caller->send(message, resHeader);
        if (err is grpc:Error) {
            log:printError("Error from Connector: " + err.message());
        }

        // Sends `completed` notification to caller.
        grpc:Error? result = caller->complete();
        if (result is grpc:Error) {
            log:printError("Error in sending completed notification to caller",
                err = result);
        }
    }
}
```

[23]

References

1. "Ballerina Language Specification" (https://ballerina.io/spec/). WSO2.
2. Open Source Contributors (18 June 2019). "Ballerina source code" (https://github.com/ballerina-platform/ballerina-lang). GitHub. {{cite web}}: |author= has generic name (help)
3. "WSO2 / LICENSE" (https://github.com/ballerina-platform/ballerina-lang/blob/master/LICENSE). *github.com*. WSO2. 2017-03-08. Retrieved 2018-03-01.

4. "Ballerina, A modern programming language focused on integration" (https://opensource.ellak.gr/wp-content/uploads/sites/5/2018/06/2018-06-Ballerina-GFOSS.pdf) (PDF): 15.
5. Jackson, Joab. "Ballerina: An API-First Programming Language" (https://thenewstack.io/ballerina-an-api-first-programming-language/). *The New Stack*. Retrieved 2018-06-11.
6. Foremski, Tom (2019-03-01). "Technology and the Arts: Celebrating Ballerina, a computer language of integration" (https://www.zdnet.com/article/ballerina-a-language-of-integration-of-technology-and-the-arts/). Retrieved 2019-07-14.
7. Lawton, George (2018-11-01). "Ballerina language promises to improve app integration" (https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Ballerina-language-promises-to-improve-app-integration). Retrieved 2019-07-23.
8. "Ballerina Microservices Programming Language: Introducing the Latest Release and "Ballerina Central" " (https://www.infoq.com/articles/ballerina-microservices-language-part-1). *InfoQ*. Retrieved 2018-06-07.
9. Earls, Alan (2019-03-01). "How does Ballerina stack up as a cloud-native programming language?" (https://searchmicroservices.techtarget.com/tip/How-does-Ballerina-stack-up-as-a-cloud-native-programming-language). Retrieved 2019-07-23.
10. Doyle, Kerry. "10 of the best programming languages to learn in 2020" (https://searchapparchitecture.techtarget.com/tip/10-of-the-best-programming-languages-to-learn-in-2020). Retrieved 2020-09-16.
11. Posta, Christian. "Evolution of Integration and Microservices with Service Mesh and Ballerina" (https://www.youtube.com/watch?v=rRrJKM0BAAo). Retrieved 2019-07-24.
12. Techworld staff. "Top programming languages you should try" (https://www.techworld.com/picture-gallery/careers/up-coming-programming-languages-for-developers-get-grips-with-3621455/). *Techworld*. Retrieved 2018-06-07.
13. Clark, James. "Ballerina Programming Language Part 0 - Context" (https://blog.jclark.com/2019/09/ballerina-programming-language-part-0.html). Retrieved 2020-09-16.
14. Clark, James. "Ballerina Programming Language Part 1 - Concepts" (https://blog.jclark.com/2019/09/ballerina-programming-language-part-1.html). Retrieved 2020-09-16.
15. "Ballerina Reinvents Cloud-Native Middleware as a Programming Language" (https://www.globenewswire.com/news-release/2019/09/10/1913510/0/en/Ballerina-Reinvents-Cloud-Native-Middleware-as-a-Programming-Language-Puts-ESB-on-the-Path-to-Extinction.html). *GlobeNewswire*. Retrieved 2020-09-16.
16. Warusawithana, Lakmal. "Rethinking Programming: The Network in the Language" (https://hackernoon.com/rethinking-programming-the-network-in-the-language-kn3z3y55). Retrieved 2020-09-16.
17. Fernando, Anjana. "Rethinking Programming: Making Sequence Diagrams Cool Again" (https://hackernoon.com/rethinking-programming-making-sequence-diagrams-cool-again-6z1p3yv9). Retrieved 2020-09-16.
18. Fernando, Anjana. "Rethinking Programming: Network Aware Type System" (https://hackernoon.com/rethinking-programming-network-aware-type-system-8o7x3yh6). Retrieved 2020-09-16.
19. Warusawithana, Lakmal. "Rethinking Programming: From Code to Cloud" (https://hackernoon.com/rethinking-programming-from-code-to-cloud-fy273yer). Retrieved 2020-09-16.
20. Fernando, Anjana. "Rethinking Programming: Automated Observability" (https://hackernoon.com/rethinking-programming-automated-observability-dn14p3yxb). Retrieved 2020-09-16.
21. Ballerina Team (16 September 2020). "Hello world service" (https://ballerina.io/learn/by-example/hello-world-service.html). ballerina.io.

22. Ballerina Team (16 September 2020). "Worker interaction" (https://github.com/ballerina-platform/ballerina-distribution/blob/master/examples/worker-interaction/worker_interaction.bal). ballerina.io.
23. Ballerina Team (16 September 2020). "gRPC unary blocking" (https://github.com/ballerina-platform/ballerina-distribution/blob/master/examples/grpc-unary-blocking/grpc_unary_blocking_service.bal). ballerina.io.

Further reading

- Fernando, Anjana, Warusawithana, Lakmal (2020) *Beginning Ballerina Programming (https://www.apress.com/gp/book/9781484251386)*, Apress (part of Springer Nature)

External links

- https://ballerina.io
- https://github.com/ballerina-platform/ballerina-lang GitHub repository.