Ballerina: a new era in programming languages

CHAKRAY

Index:

1. Introduction

2. What makes Ballerina different?

3. Ballerina, the added value of integration

4. Ballerina Tools: programming tools

5. Conclusions

Img: Freepik

1. Introduction

WSO2 has proven to be a leading, groundbreaking company that is always ready to grow and offer new solutions. More than 3 years ago, WSO2 started working on Ballerina. This is a new programming language designed to be the best when writing that need to talk and coordinate over a network. Since the launch of Ballerina in 2018, it has received several recognitions, its use has become significantly widespread, and nowadays there are already several companies that have adopted it for their service-based architectures. Sanjiva Weerawarana, CEO and Chief Architect of WSO2, explained that the goal of his company is to implement applications conceived to be used by both people and organizations of all kinds.

The driving force of Ballerina in WSO2 is that it is the only fully integrated enterprise platform that allows companies to create, integrate, manage, protect and analyze their on-site, cloud, mobile device and Internet of Things APIs, applications and Web services.

> *“WSO2 is that it is the only fully integrated enterprise platform that allows companies to create, integrate, manage, protect and analyze their on-site, cloud, mobile device and Internet of Things APIs, applications and Web services.”*

Companies in countless sectors (finance, logistics, telecommunications, health, etc.) have acknowledged the debut of a programming language that is expected to become the new benchmark. Mostly, due to the connection between services and applications that it offers in all kinds of integration scenarios.

Ballerina significantly improves the developers’ productivity, turning API iterations into a fast and agile process. It provides a native ‘API Gateway’ that is ready to connect any Ballerina-managed service to an API management solution, or act as a micro-gateway. In essence, both the language and the compiler are prepared to operate in a distributed environment, where the artifacts created by the development team can be integrated without effort.

What makes Ballerina different?

Ballerina was born as an internal WSO2 project – a company that has made significant contributions to the world of integration in the enterprise landscape. WSO2 engineers (like many others in cloud) felt increasingly frustrated by the fact that most enterprise buses (ESBs) required XML as a standard format when configuring message flows – which completely breaks the developers' routines. Developers demanded a more natural and simple way to build drivers and encapsulate the logic to handle incoming requests. Ballerina is highly visual, and removed from code language that is usually hard to understand by a non-expert audience, therefore allowing it to be showcased to third-parties.

> *"Ballerina is a visual language that is easy to understand even by non-experts."*

But what makes Ballerina different from any other programming language known so far?

Flexibility

By using sequence diagrams as a starting point, Ballerina can generate integrations. It is ready to employ plug-ins in Ballerina code in IntelliJ IDEA, Vim and others. It also allows code to be written in Ballerina itself or in Swagger. A good example of this flexibility is that the community of users that contributing to its codebase grows by the day. Its application programming interface (API) also allows it to leverage the functionality of other programs.

Intuitive Visualisation

The semantics of the Ballerina language itself was designed by modelling them as independent parts that interact with each other by means of structured interactions. Thanks to this, the flow all Ballerina programs can be visualized as sequence diagrams with endpoints, including synchronous or asynchronous calls.

Ballerina Composer is an integrated tool that allows for the creation of Ballerina services through sequence diagrams. This means that developers can apply best practices in strong integrations in a more natural way.

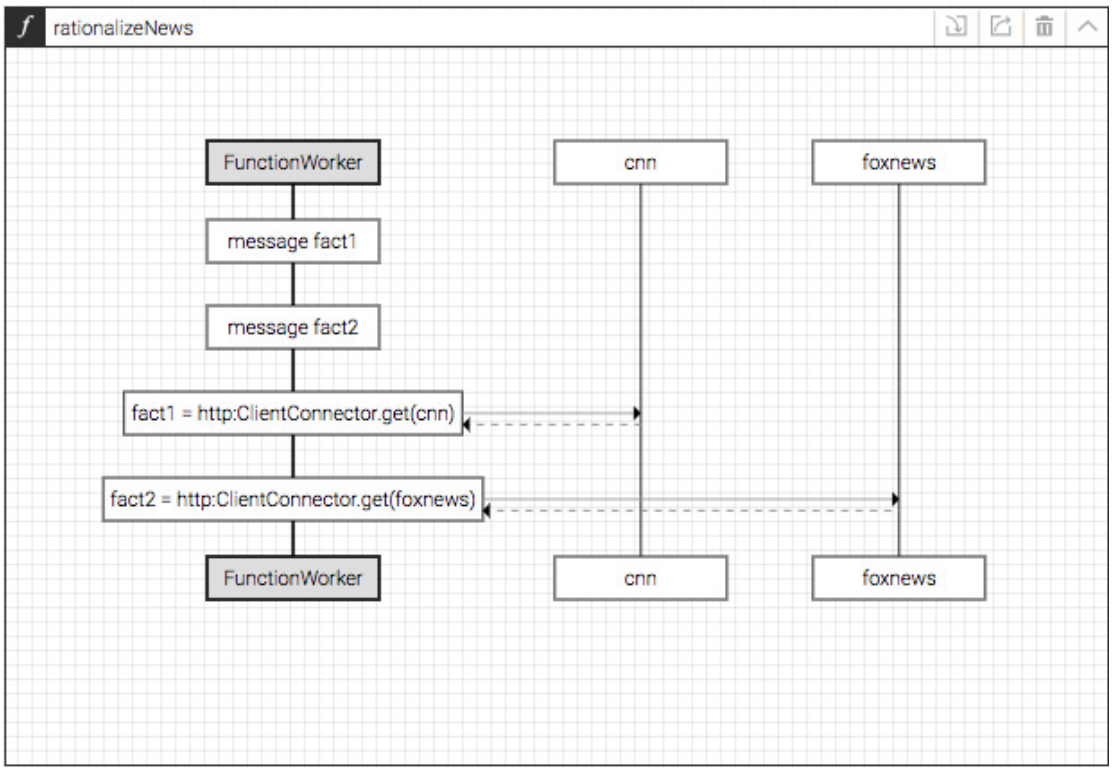
WSO2's efforts to make it visual and intuitive has borne fruit, because it is, more than anything, a graphical language that can be understood by a wide spectrum of professionals, even if they are non-experts. This also simplifies the task of explaining things to a group. By doing something as simple as dragging & dropping elements on one same image, a diagram is generated as the code of the integration is created. What sets **Ballerina** apart is that its displays both a text and graphical syntax with the same expressive capabilities,

which are also fully reversible. Syntax exceptions are eliminated and everything results from intuitive and key concepts of the language. Furthermore, Ballerina follows Java and Go to provide a platform-independent programming model that abstracts programmers drom specific details.

Based on the idea of a “main block” in the code, most languages do not offer a good model to support concurrency, which is absolutely essential for software that interacts with other software. Some current-day languages (Go) do offer native concurrency models, but they are not easy to master.

Ballerina is neither an object-oriented language nor a functional language, even though it supports both objects and lambdas. It is a traditional declarative language, much like C. If the history of distributed programming has taught us anything through CORBA and JavaBeans is that most artifacts travelling over the network are not objects.

Its ease of use makes it possible to draw both the specific integration scenario and the components that must interact with each other. Introducing an integration logic is not complicated. And, how can the initial view be recovered? It is as simple as switching to the same origin view.



Ballerina diagram screenshot

Constant development

Ballerina was introduced last February as a new product, but it had already been (and still is) an ongoing project. The Ballerina language is now ready for production, even though release 1.0 has not yet been closed. This is because work is being done to offer long-term backward compatibility since that moment, which makes it very important for the language to have an extremely high level of stability.

Various tests have demonstrated that HTTP services in a manager/worker model (a standard and widely-used model in concurrent programming) offer a performance of 20000-25000 transactions per second on a basic hardware. These numbers are already 20% over those of Spring Boot and 50 percent over similar solutions on Python, as revealed by Tyler Jewel, from WSO2.

The next steps for Ballerina are focused on optimizing the capabilities for stateful services, serverless execution, complex multilayer compensations and optimizations for large-scale implementation.

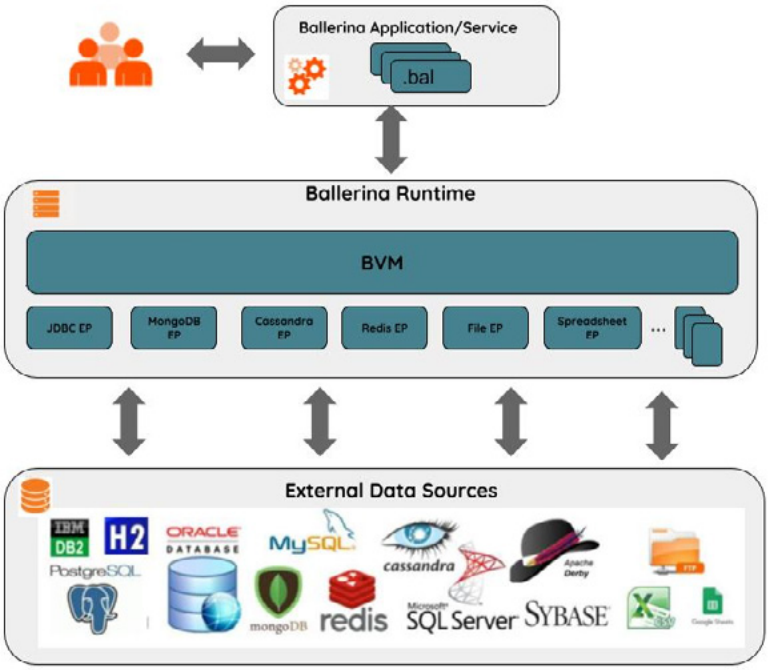
3. Ballerina, the added value of integration

Integrating all the systems that a company works with and automating the communication between them, even when they speak different languages, is one of the keys toward improving its efficiency. This kind of integration is possible, without this implying any complexity and **regardless of whether those systems are located at different locations,** such as on-site, on the cloud or on other devices.

> *“Ballerina is capable of handling programs that go from the simplest to the most complex.”*

Ballerina has been created for integration to be one of its strong suits. This translates into the fact that this advanced language is capable of handling programs that go from the simplest to the most complex. The latter include, for example, content-based routing scenarios or complex service chaining. At the time, Ballerina offers native support for **Swagger, REST, XML and JSON, as well as data and mapping tables. And also connectors, both for Web APIs and non-HTTP APIs.** And all of this, right off the bat.

Ballerina possesses a rich packet ecosystem that connects to different types of data sources, SQL and noSQL databases, CSV files, FTP files, Google Spreadsheets, etc. For each of these data sources there is a specific client, which runs on Ballerina VM.



WSO2's product also has **social network service connectors**, including Facebook and Twitter. In addition, its high speed offers much shorter execution times than other languages, which is always welcome by users.

Ballerina supports high-performance implementations, including micro-services and micro-integrations, which are increasingly seeing used in digital products. it does this with a low latency, limited memory use and an agile implementation. With its integration and automation among different services, **the company's information can achieve** the required **operational fluidity.**

4. Ballerina Tools: programming tools

Ballerina offers a full set of development tools – one of its competitive advantages when compared to other programming languages. This makes work much easier.

Composer

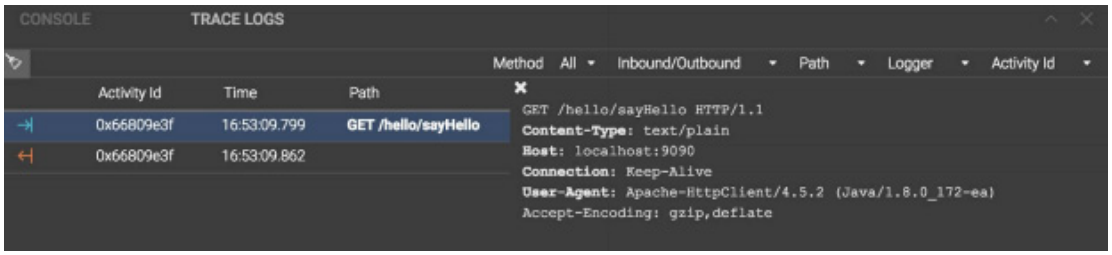
Composer is an independent IDE that can be installed with Ballerina's installer. Composer has features that make it quite interesting and improved, and even with advantages that other highly sophisticated editors lack. In summary, it is the main editor to develop Ballerina programs. Its main features are:

- It facilitates **element programming** and the **development of compositions** with ease, through a simple drag & drop gesture.
- It can **execute and debug Ballerina programs** directly from the editor itself.
- It allows for the **editing of any interface.** It leverages the advantages of an open-source framework that is supported by a great ecosystem of tools such as Swagger.

In addition, Composer allows for switching between different visualizations, such as the **Design view, the Source view and the Swagger view**, as well as for working in your preferred one. While working in one of the views – whether the code view or the diagrams – the rest of the views are updated, which makes it possible to switch views as much as you need. The Swagger view is only for 'defining services', which means that it cannot be used in other cases (creation of an executable program, etc.).

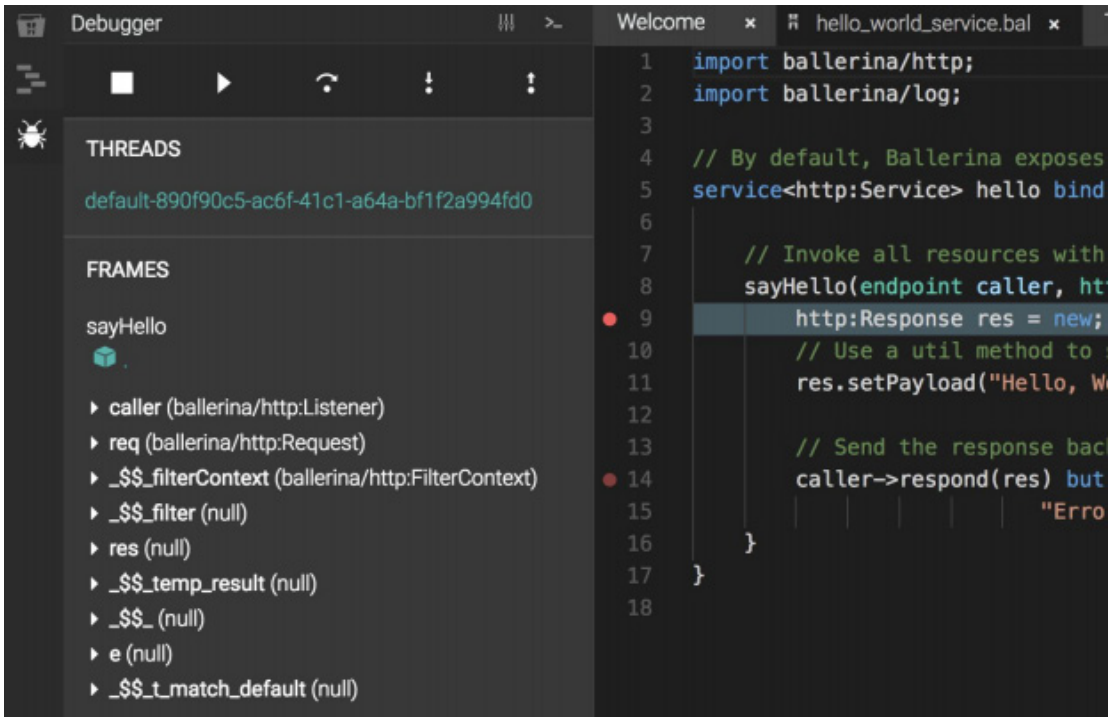
Composer offers the possibility of debugging by simply activating this view from Composer and setting up the breakpoints in the source. This is one of its most demanded functionalities during the first stages of development of Ballerina, and is fully operational in the current version.

“Ballerina offers a full set of development tools – one of its competitive advantages when compared to other programming languages.”

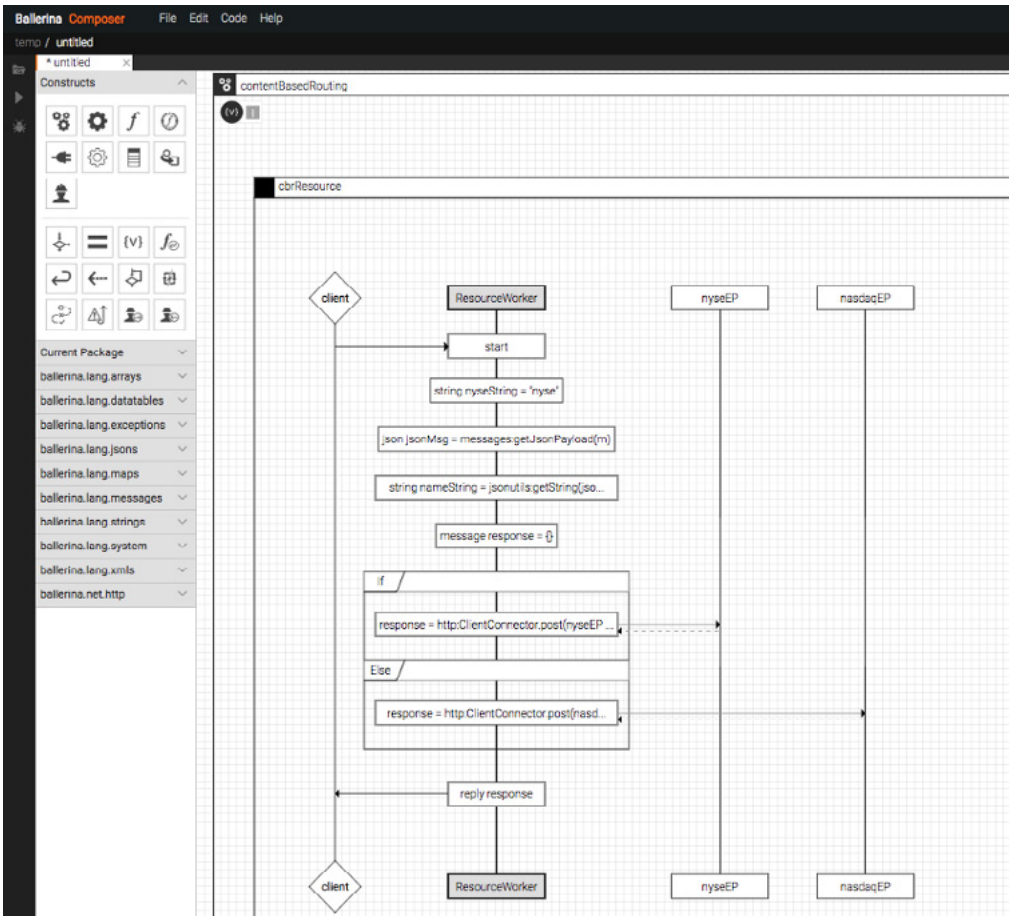


Another good thing about Composer is that it allows for services to be invoked through the 'Try It' option in the debug panel. This allows you to selectively execute invocations of the various integration points, and quickly model their behavior.

Each time a service is invoked, the tool offers an overview of the full trace logs for the invoked services, which can be accessed from the 'Trace Log' view.



Ballerina offers a full set of development tools – one of its competitive advantages when compared to other programming languages.



Screenshot of Ballerina Composer.

Testerina

Testerina is the system testing unit for Ballerina programs. It allows users to emulate Ballerina programs within a test environment. Users themselves can also write in unit tests to test the source code using this framework. Test results can be printed using Testerina.

Like any state-of-the-art unit testing framework, it allows for dependencies to be defined between tests, service mocking and the ability to group logical units.

Docerina

Docerina is used to generate API documentation in the Ballerina language. The format in which this API documentation is generated is HTML, even though this can be expanded to support as many additional formats as necessary.

There are several style themes for the HTML documents. In addition to those provided by default, you have the option of using a great variety of themes developed by the community.

Connectors

With Connectors, the client can establish contact with the various cloud systems and with different APIs. Therefore, this offers the advantage for users to be able to write their own connectors in the Ballerina language and used them within any other program. It is one of the extensions of Ballerina that are, in practice, most interesting.

In Ballerina it is possible to define a set of actions using connectors:

- *setKeyValue :* Set key/value.
- *getValue :* Obtain the value for a given key.
- *updateValue :* Update the value for a given key.
- *deleteKey :* Delete a key.
- *createDir :* Create a directory.
- *listDir :* List a directory.
- *deleteDir :* Delete a directory.

The increasing popularity of the 'Ballerina central' service is worth mentioning. This is a marketplace where connectors are published for a myriad of services and that, naturally, have a version control system and a popularity ranking. This model follows WSO2's 100% open-source philosophy, where an increasing number of developers and companies are sharing their connectivity solutions for standard (and non-standard) services.

Editor plugins

These are plugins as popular as **Atom, Intellij Idea, Vim, Sublime and VSCode** – these are some of the most widespread plugins used by Ballerina. They are plugin tools for source code editors that are quite useful.

Container support

Container support is conceived for the simple implementation of programs for Ballerina packaging with Docker. Uses the required command in each case. It is a tool that makes implementation much easier.

At the moment of installation, the compiler generates the BVM instructions in libraries (.balo) or in binary format (.balx)

It is also possible to generate Docker images directly through extensible notations, allowing their use in Docker and k8s architectures.

5. Conclusions

Ballerina has significantly and rapidly evolved since its conception, nowadays being a solution sought by corporate early adopters. The prioritization of Ballerina's roadmap has been shaped by the feedback of the engineers who used it first hand, which has led it to experience an increase in capabilities across different landscapes during recent years.

The current Ballerina version is pre-1.0. However, the final 1.0 version is close, with v0.990.0 being a strong candidate for 1.0. The team has anticipated that, since 1.0, changes in the language will be fully backward-compatible.

We might see a Ballerina 2.0 specification in the distant future, through the team has already stated that their efforts since release data will focus on the ecosystem around Ballerina more than on the language itself. This reflects Ballerina's goal of becoming a benchmark language with a high, long-term stability.

This is something groundbreaking that will **close the age-old gap between programmers and all other professionals**. Many sectors will be able to integrate their processes in a clear and intuitive manner, to the point that old codes for experts may start to lose ground.

This product by WSO2 guarantees fluidity **(speed and low latency) without requiring a significant amount of memory.** Right off the assembly line, it offers data tables and mapping, as well as native support for frameworks such as Swagger, REST, XML and JSON. And also connectors for the application programming interface from the Web and for non-HTTP APIs.

Ballerina can be downloaded from its official website. There it is possible to download the 'Ballerina runtime' package, which only contains the runtime environment required to execute the Ballerina program, or the full package. The latter is the most appropriate choice for anyone wishing to enjoy all the tools, and includes **Composer** for visual editing, **Docerina** to generate API documentation for Ballerina programs, and **Testerina**, the system's testing unit to test the code and even obtain results from those tests in print.

¡Thank you!

About Chakray:

We do things well

At Chakray we specialize in architecture, consulting and Critical Information System training services. Our team develops Open Source projects with the most innovative software from WSO2, with a single purpose: to make the technology of your company the best out there.

Do you want to improve your systems? Consult with our experts!

Ask our consultants, no strings attached
We will help you find the best solution for your project.

CONTACT US

info@chakray.com | www.chakray.com