



Level Up Your APIs: GraphQL with Ballerina

Thisaru Guruge

April 2024

# Housekeeping

- Please ensure your mic is muted during the presentation and demo
- Please use the chat box to ask any questions you may have
- We will have a live Q&A at the end of the session and take your questions during this time
- This session will be recorded

# In this presentation

Understanding GraphQL; What, Why, and When

---

Simplify Your GraphQL Development in Ballerina

---

Best Practices for Maintainable GraphQL APIs

---

Q&A

---

# Understanding GraphQL; What, Why, and When

- What?
  - A query language and a runtime for APIs
  - A protocol for communicating between two endpoints
  - Defines the API using a GraphQL schema
  - Queries, Mutations, and Subscriptions

# Understanding GraphQL; What, Why, and When

- Why?
  - To overcome problems like overfetching, underfetching
  - Provide more power to the clients
  - Provide robust and typesafe APIs

# Understanding GraphQL; What, Why, and When

- Questions
  - Is GraphQL going to replace REST?
  - Is it okay to replace every API into a GraphQL API?
  - When to use GraphQL?
- When?
  - Complex systems with inter-related data
  - Flexibility in client data requirements

# Simplify Your GraphQL Development with Ballerina

- **Ballerina and GraphQL**
  - Similarity between the type systems
  - Data oriented nature of Ballerina
  - Type safety and concurrency safety
  - Built-in network abstractions

# Simplify Your GraphQL Development with Ballerina

- Develop GraphQL in Ballerina
  - GraphQL Visual Designer and GraphQL CLI
  - Supports both code-first and schema-first approaches
  - Out of the box support for subscriptions
  - Built-in GraphQL client
  - DataLoader and server-side caching for optimize the API
  - Comprehensive security features



# Sample Service

```
import ballerina/graphql;

service /graphql on new graphql:Listener(9090) {
    resource function get greeting() returns string {
        return "Hello, World!";
    }
}
```

# GraphQL Best Practices

- Avoid versioning
- Maintain nullability
- Batching and caching; avoid N+1 problem
- GraphQL federation

# Check out the sample

[GitHub](#)

## Ballerina Community



**Discord** : <https://discord.gg/ballerinalang>



**SO**: <https://stackoverflow.com/questions/tagged/ballerina>



**Twitter**: <https://twitter.com/ballerinalang>



**GitHub** : <https://github.com/ballerina-platform>

# Questions?

