

# Making A Simple API project using GrpahQL with .Net 7

Presented By : Yin Win Phyu

# Table of contents

01

About GraphQL

02

Navigating API Choices

03

Types of GraphQL

04

The Easy and Simple Library  
for GraphQL

05

Simple API Project By  
GraphQL



01

# About GraphQL



# About GraphQL

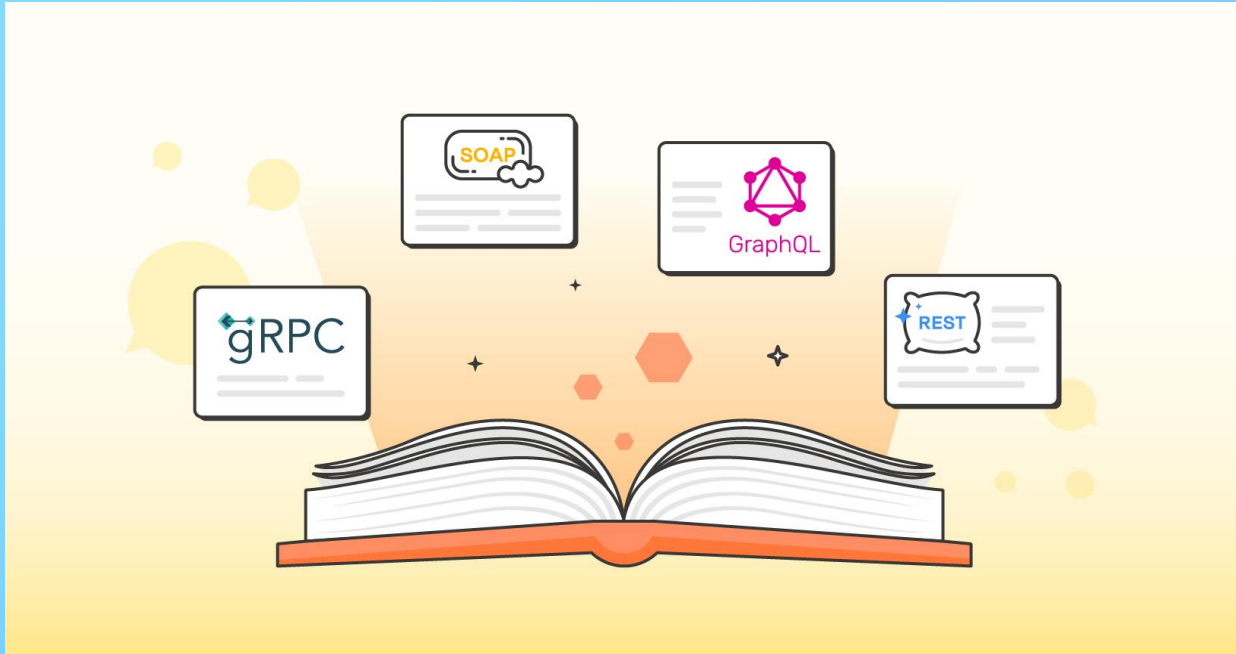
GraphQL is a query language for APIs and a runtime for executing those queries by using a type system you define for your data. It provides a more efficient, powerful, and flexible alternative to REST for accessing data in an API.

02

# Navigating API Choices



# Navigating API Choices



# 1. GraphQL



## Overview

- **GraphQL** is a query language for APIs

## Key Features

- **Flexible Queries:** Clients specify exactly what data they need.
- **Single Endpoint:** Uses a single endpoint to handle all requests.
- **Strongly Typed Schema:** Schema defines types and relationships.
- **Introspective:** Clients can query the schema itself.
- **Real-time Capabilities:** Supports subscriptions for real-time updates.



# 1. GraphQL



## Pros

- Efficient data fetching: Only the data requested by the client is returned.
- Strongly typed schema ensures valid queries.
- Reduces over-fetching and under-fetching of data.

## Cons

- Can be complex to implement and optimize.
- Requires learning a new query language.
- May lead to deeply nested queries that are hard to optimize.



## 2. REST API



### Overview

- architectural style for designing networked applications.
- Uses standard HTTP methods

### Key Features

- **Resource-Based:** Everything is a resource accessible via a unique URL.
- **Stateless:** Each request from the client contains all the information needed to process the request.
- **Cacheable:** Responses must define themselves as cacheable or non-cacheable.

## 2. REST API



### Pros

- Simple and widely understood.
- Leverages existing HTTP infrastructure.
- Easy to test with standard tools like Postman.

### Cons

- Can lead to over-fetching or under-fetching of data.
- Multiple endpoints can lead to complex client logic.
- Lacks a standard way to describe APIs (though OpenAPI helps).

# 3. SOAP

## Overview

- a protocol for exchanging structured information in web services.
- Uses XML for message

## Key Features

- **Extensive Standards:** Built-in error handling, security (WS-Security), and transaction compliance.
- **Strong Typing:** Uses WSDL (Web Services Description Language) to describe services.
- **Stateful Operations:** Supports complex operations and stateful interactions.



# 3. SOAP



## Pros

- Highly extensible and standardized.
- Built-in error handling and security features.
- Suitable for enterprise-level applications requiring high security.

## Cons

- Verbose XML format increases bandwidth usage.
- Steeper learning curve and complexity.
- Can be slower due to larger message sizes.

## 4. gRPC



### Overview

- open-source remote procedure call system developed by Google.
- Uses Protocol Buffers (protobuf) as the interface definition language.

### Key Features

- **High Performance:** Efficient binary serialization.
- **Bi-Directional Streaming:** Supports client, server, and bi-directional streaming.
- **Strongly Typed:** Interface defined using Protocol Buffers.
- **Supports Multiple Languages:** Provides language-specific bindings.

## 4. gRPC



### Pros

- High performance and low latency.
- Strongly typed contracts with Protocol Buffers.
- Supports advanced use cases like bi-directional streaming.

### Cons

- Requires additional tooling (e.g., protoc compiler for protobuf).
- Not as human-readable as JSON (used in REST).
- Learning curve for Protocol Buffers and gRPC concepts.

03

# Types of GraphQL





# Types of GraphQL

## Schema Definition

```
1  Run > schema {  
2    query: BlogQuery  
3    mutation: BlogMutation  
4  }  
5  
6  Run > type BlogQuery {  
7    blog(pageNo: Int!, pageSize: Int!): [BlogModel!]!  
8    blogById(id: Int!): BlogModel!  
9    allBlog: [BlogModel!]!  
10 }  
11  
12 Run > type BlogMutation {  
13   createBlog(blog: BlogModelInput!): BlogModel!  
14   updateBlog(blog: BlogModelInput!): BlogModel!  
15   deleteBlog(id: Int!): Boolean!  
16 }  
17  
18 Run > input BlogModelInput {  
19   blogId: Int  
20   blogTitle: String  
21   blogAuthor: String  
22   blogContent: String  
23 }  
24  
25 Run > type BlogModel {  
26   blogId: Int  
27   blogTitle: String  
28   blogAuthor: String  
29   blogContent: String  
30 }  
31
```

04

# The Easy and Simple Library for GraphQL



# The Easy and Simple Library for GraphQL



## Hot Chocolate

- is a .NET GraphQL server library
- allows you to create a GraphQL API quickly and efficiently
- provides tools for building robust GraphQL schemas and resolving queries and mutations, simplifying the process of integrating GraphQL into .NET applications.

## Banana Cake Pop

- is a GraphQL IDE that comes with the Hot Chocolate framework.
- provides a user-friendly interface for exploring and testing your GraphQL APIs, similar to GraphQL Playground or GraphiQL, but specifically tailored for use with Hot Chocolate.









05

# Simple API Project By GraphQL






# Simple API Project By GraphQL

The Libraires that need to install not using Hot Chocolate

 <b>GraphQL.MicrosoftDI</b> by GraphQL.MicrosoftDI Microsoft DI extensions for GraphQL.NET	7.8.0
 <b>GraphQL.Server.Authorization.AspNetCore</b> by Shane Krueger, Pekka Heikura, Ivan Maximov, et al. Integration of GraphQL.NET validation subsystem into ASP.NET Core	7.7.1
 <b>GraphQL.Server.Transports.AspNetCore</b> by Shane Krueger, Pekka Heikura, Ivan Maximov, et al. HTTP middleware for GraphQL	7.7.1
 <b>GraphQL.Server.Ui.Altair</b> by Shane Krueger, Pekka Heikura, Ivan Maximov, et al. GraphQL Altair integration for ASP.NET Core	7.7.1
 <b>GraphQL.Server.Ui.Playground</b> by Shane Krueger, Pekka Heikura, Ivan Maximov, et al. GraphQL Playground integration for ASP.NET Core	7.7.1
 <b>GraphQL.SystemTextJson</b> by GraphQL.SystemTextJson System.Text.Json serializer for GraphQL.NET	7.7.1 7.8.0
 <b>Microsoft.VisualStudio.Azure.Containers.Tools.Targets</b> by Microsoft Targets files to enable the Visual Studio Tools for Containers.	1.19.6 1.20.1
 <b>Swashbuckle.AspNetCore</b> by domaindrivendev Swagger tools for documenting APIs built on ASP.NET Core	6.6.2

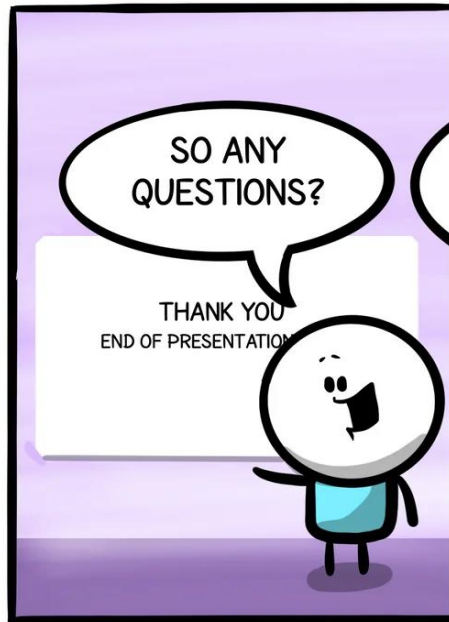
The Libraires that need to install using Hot Chocolate

 <b>HotChocolate.AspNetCore</b> by ChilliCream authors and contributors This package contains the GraphQL ASP.NET Core middleware for Hot Chocolate. Moreover, this package includes the Banana Cake Pop middleware, which provides you with our beloved GraphQL IDE middleware.	13.9.5
 <b>Microsoft.EntityFrameworkCore</b> by Microsoft Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.	7.0.0 8.0.6
 <b>Microsoft.EntityFrameworkCore.SqlServer</b> by Microsoft Microsoft SQL Server database provider for Entity Framework Core.	7.0.0 8.0.6

# Simple API Project By GraphQL

## Showing Code Time

WHAT I SAY



WHAT I THINK



[workchronicles.com](http://workchronicles.com)

follow on Instagram / Twitter / Facebook