GraphQL: A Beginner's Guide

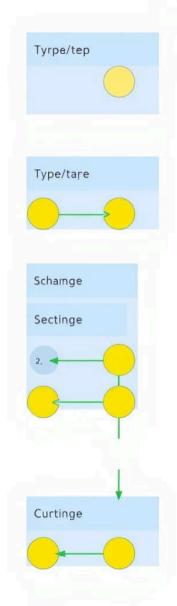
GraphQL is a modern query language for APIs. It helps clients request exactly the data they need.

This presentation will introduce GraphQL basics for new developers.





Graph.fpcl



Core Concepts: Schemas & Types

Schema

Defines data structure as a contract between client and server.

Types

- Scalar: Int, String, Boolean
- Objects, Lists, Non-Null

Example

type User { id: ID!, name: String, email: String }

Queries: Asking for Specific Data

Purpose

Request only the data fields you need from the server.

Example

{ user(id: "123") { name, email } }

Benefits

- Avoids over-fetching
- Prevents under-fetching



Mutations: Modifying Data

1 2

Create Update Delete

Add new data entries. Modify existing data. Remove data securely.

Example: mutation { createUser(name: "John", email: "john@example.com") { id name email } }

Made with **GAMMA**

Resolvers: Connecting Schema to Data

Function Role

Fetches data for each schema field.

Integration

Bridges GraphQL and databases or APIs.

Example Resolver

(user) => user.getName()

```
faciclatios) - resolert up(igpcolale, atapil)
           ior(ceristald] in_sterleage]
 product: -tricignottom> )
 etert = fele(Car"eas)))
immoltle: apess ertfort-= refien, GrapOl - traint: Lamne-ints (peacing, Lesset courting) to - traints (peacing) - fittiant)
      Certiecler: recianecinat-s))
      wertie-tiperil/swwcler ipcf.(ody)
       resolver "ptafeharlals. ("more "resolver")
       sence= Cparice. Alternative
        thertel setils repolator: "Backler, 2000)
     eraphating casuel: (rlerging: (nection:
   dortiaction fewsitalaye: (noselor Ints(Jatel))
   - Decicle arcuial, inscrialt, Cursec Onccubater thouth, satural
       bewict -wallicatter(((benceclor "sellar)
          recommecior titer" asad)
          prooftraction: Inftinatate for thesechespoints to
          proctiaction( fore"incephre_lnsgrelstactor.
         yttalavsotion!)
         practiracting: = retional -Teromating/incetting/
          reportion, orf iresister = "Danplansties!!!
                        //: (earchtenc(asactertiess)
          -mtrisection-Ontenns(.t. = fart)
              processe: (lrapicles intectal, )
                        (fflowarriation traf- Geral
                                            Made with GAMMA
```

REST API vs. GraphQL: Key Differences

REST API

- Multiple endpoints
- Fixed data structures
- Over- and under-fetching
- Versioning challenges

GraphQL

- Single endpoint
- Flexible queries
- Schema driven
- Improved efficiency

GraphQL Use Cases: When to Use It

Complex Data Needs

Ideal for multiple data sources and complex queries.

Mobile Apps

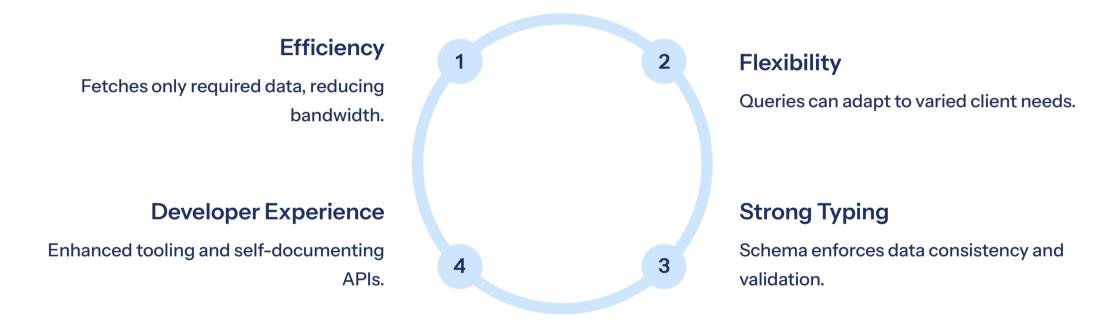
Optimizes data fetching and network usage.



- Netflix Ul
- GitHub Public API



Summary: Benefits of GraphQL



GraphQL is a powerful choice for modern, complex APIs.