BTK AKADEMİ

Web Servislerine Giriş

Zafer CÖMERT

BTK AKADEMİ

Bölüm 7

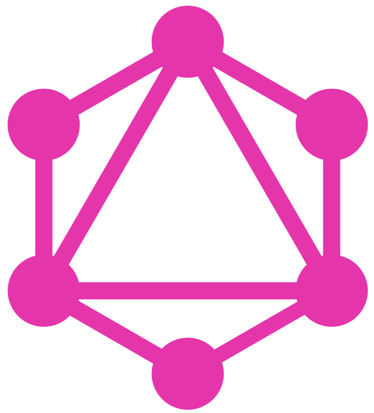
GraphQL

Web Servislerine Giriş

BTK AKADEMİ

GraphQL

1. Temeller
2. Kökeni
3. Kullanım Örnekleri
4. Spesifikasyon
5. GrapQL İletişimi
6. Şemalar
7. Sorgular
8. GraphQL Çözdüğü Sorunlar
9. GraphQL vs REST API
10. Uygulama Örneği



GraphQL

GraphQL, açık kaynaklı bir veri sorgulama ve manipülasyon dilidir.

BTK AKADEMİ

GraphQL

- GraphQL, açık kaynaklı bir veri sorgulama ve manipülasyon dilidir ve uygulama API'ler için kullanılır.

- API'ler, iki uygulamanın belirli bir kurallar setini takip ederek istekler ve yanıtlar biçiminde bilgi değiş tokuş etmelerine izin verir.

- GraphQL, bir API tüketicisinin gereksiz bilgi almaksızın bir uygulama sunucusundan belirli verileri istemesine izin verir.

GraphQL

- Geleneksel REST API mimarilerinde sabit bir veri yapısı sağlanır.
- İstemcilerin ihtiyaç duymadıkları gereksiz bilgileri filtrelerler.
- Güvenlik açısından, GraphQL'in tasarımı bir avantaj sağlar. Çünkü GraphQL, istemcinin açıkça istemediği verileri döndürmez, bu da bilgi sızıntısı sorunları için riskleri azaltır.

BTK AKADEMİ

GraphQL, bir iletişim örüntüsüdür.

GraphQL’in Kökeni

Facebook tarafından geliştirilmiştir.

GraphQL'in Kökeni

- GraphQL, Facebook tarafından 2012 yılında geliştirilmeye başlandı ve birkaç yıl boyunca kendi üretim ortamlarında kullanıldıktan sonra 2015 yılında açık kaynak yazılım olarak yayımlanmadan önce içeride geliştirilen bir teknolojiydi.

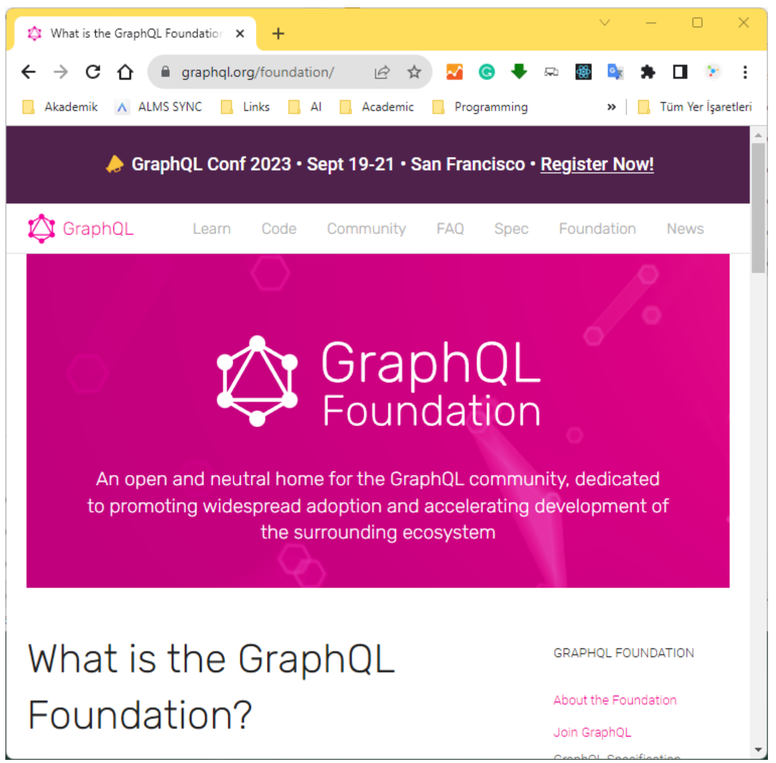
- Aynı yıl içerisinde, Facebook GraphQL spesifikasyonunu ve JavaScript kullanılarak geliştirilmiş bir referans uygulama olan GraphQL.js'i geliştirip yayınladı.

https://github.com/graphql/graphql-js

GraphQL'in Kökeni

- Günümüzde GraphQL, küresel teknoloji şirketleri tarafından kurulan GraphQL Foundation tarafından sürdürülmektedir.

- Bu kuruluş, GraphQL'i sürdürmek için mentorluk ve proje hibeleri sağlar, GraphQL ticari markası politikalarını yönetir, projeler için hukuki destek sunar ve toplulukla ilgili altyapıyı destekler.



https://graphql.org/foundation/

BTK AKADEMİ

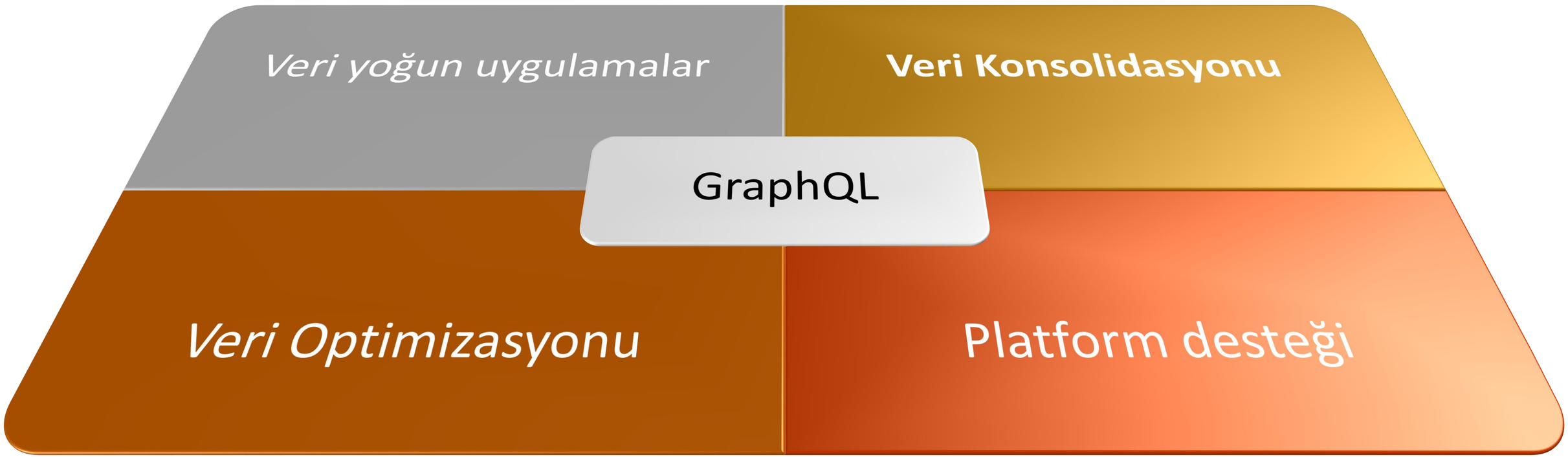
GraphQL, Facebook'un kendi ihtiyaçları için geliştirdiği bir teknolojiden, açık kaynak bir standart haline gelmiş ve küresel bir topluluk tarafından desteklenen bir veri sorgulama ve manipülasyon dili haline gelmiştir.

BTK AKADEMİ

GraphQL Kullanım Örnekleri

GraphQL bir iletişim modeli olduğundan, çalışmaya başlamanıza yardımcı olacak ve her türlü dilde GraphQL'i destekleyen birçok araç vardır.

GraphQL Kullanım Örnekleri



BTK AKADEMİ

GraphQL, farklı uygulama senaryolarında ve platformlarda veri sorgulama ve iletişim ihtiyaçlarını karşılamak için kullanılabilir ve özellikle yoğun veri içeren, veri konsolidasyonu gerektiren ve bant genişliği veya performans optimizasyonu önemli olan uygulamalarda tercih edilmektedir.

BTK AKADEMİ

GraphQL Spesifikasyonu

GraphQL spesifikasyonu, 2015 yılında Facebook tarafından kamuya açık olarak yayınlanmış bir belgedir.

GraphQL Spesifikasyonu

- GraphQL spesifikasyonu, 2015 yılında Facebook tarafından kamuya açık olarak yayınlanmış bir belgedir.

- Bu belge, GraphQL'in tüm uygulamalarının uyması gereken kuralları, tasarım prensiplerini ve standart uygulamaları tanımlar.

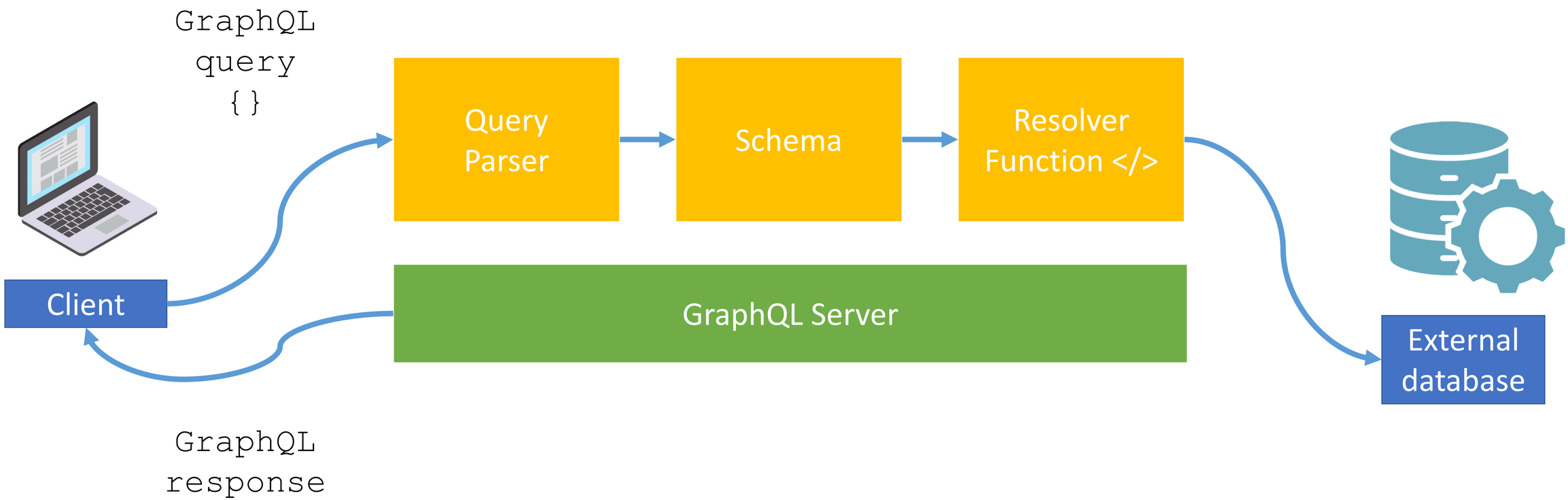
- GraphQL'in farklı programlama dilleri için uygulanmasına rehberlik eder.

BTK AKADEMİ

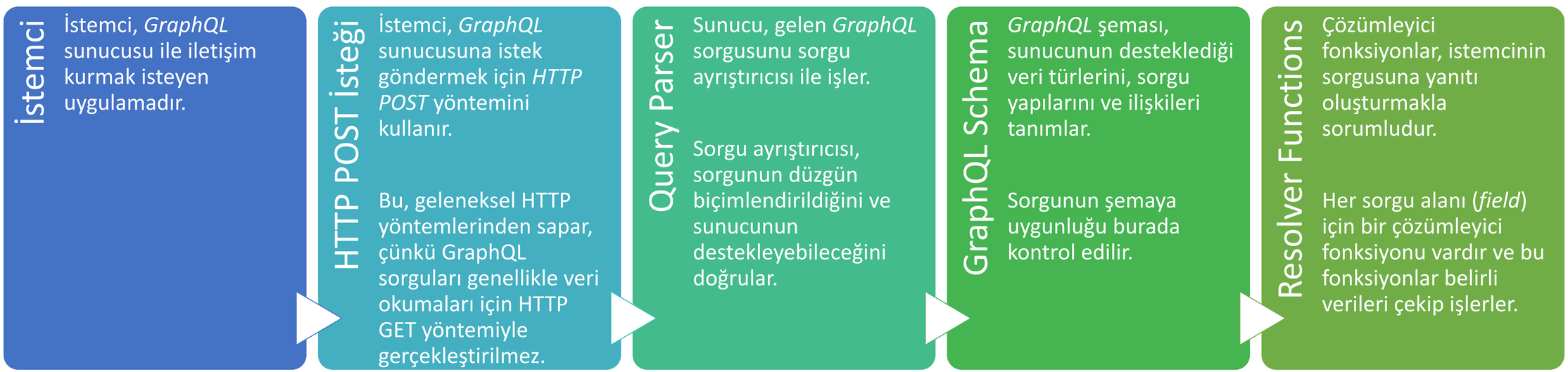
GraphQL İletişimi

GraphQL bir iletişim desenidir.

GraphQL İletişimi



GraphQL İletişimi



BTK AKADEMİ

GraphQL Şema

GraphQL Schema

GraphQL Şema

- GraphQL şemaları, istemcinin sorgulayabileceği veri türlerini ve bu verilerin nasıl ilişkilendirileceğini tanımlar.

- Şemalar, Schema Definition Language (SDL) adı verilen bir dille tanımlanır ve temel olarak iki tür veri yapısını içerir.

Object types

Scalar types

BTK AKADEMİ

GraphQL Şema

```
1  const UserType = new GraphQLObjectType({
2      name: 'User',
3      fields: () => ({
4          id: { type: GraphQLID },
5          firstname: { type: GraphQLString },
6          lastname: { type: GraphQLString },
7          salary: { type: GraphQLInt }
8      })
9  });
```

GraphQL Şema

```
const RootQuery = new GraphQLObjectType({
    name: 'RootQueryType',
    fields: {
        user: {
            type: UserType,
            args: { id: { type: GraphQLID } },
            resolve(parent, args) {
                // code to get data from db / other source
                console.log("user id", args.id)
                return users.find(u => u.id === args.id)
            }
        },
        users: {
            type: GraphQLList(UserType),
            resolve(parent, args) {
                return users;
            }
        },

    }
})
```

BTK AKADEMİ

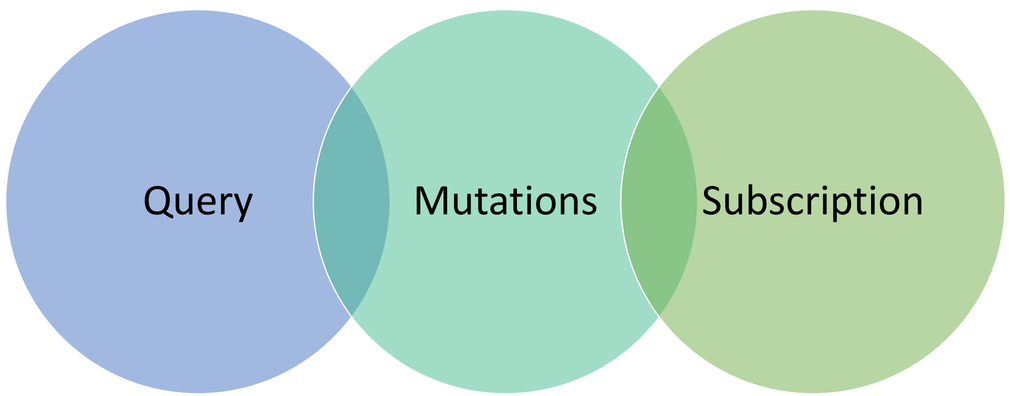
GraphQL Sorgular

GraphQL Query

GraphQL Sorgular

- GraphQL sorguları, istemcilerin belirli verileri sorgulamak için kullanabileceği deklaratif bir GraphQL sorgu dilinde yazılmış özel sorguları içerir.

- GraphQL'de tüm sorgular, operasyonun kök türünün tanımıyla başlar.



BTK AKADEMİ

```
{
  users {
    id
    firstname
    lastname
  }
}
```



```
{
  "data": {
    "users": [
      {
        "id": "1",
        "firstname": "Zafer",
        "lastname": "Cömert"
      },
      {
        "id": "2",
        "firstname": "Can",
        "lastname": "Umut"
      },
      {
        "id": "3",
        "firstname": "Yücel",
        "lastname": "Ufuk"
      },
      {
        "id": "4",
        "firstname": "Filiz",
        "lastname": "Derya"
      }
    ]
  }
}
```

```
{
  users {
    id
    firstname
    lastname
    salary
  }
}
```



```
{
  "data": {
    "users": [
      {
        "id": "1",
        "firstname": "Zafer",
        "lastname": "Cömert",
        "salary": 35
      },
      {
        "id": "2",
        "firstname": "Can",
        "lastname": "Umut",
        "salary": 40
      },
      {
        "id": "3",
        "firstname": "Yücel",
        "lastname": "Ufuk",
        "salary": 45
      },
      {
        "id": "4",
        "firstname": "Filiz",
        "lastname": "Derya",
        "salary": 15
      }
    ]
  }
}
```

BTK AKADEMİ

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

GraphQL Query Parser and Resolver Operations

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

- GraphQL sunucusu, bir sorgu aldığında bu sorguyu işlemek için bir sorgu ayrıştırıcı (query parser) kullanır.

- Sorgu ayrıştırıcısı, gelen sorgu dizesini soyut sözdizimi ağacına (abstract syntax tree - AST) dönüştürmek ve şemaya karşı doğrulamakla sorumludur.

- AST, sorguyu temsil eden hiyerarşik bir nesnedir. Bu nesne, alanlar, argümanlar ve diğer bilgileri içerir ve farklı dil çözümleyicileri tarafından kolayca gezilebilir.

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

- GraphQL güçlü bir şekilde tiplendirilmiştir, yani istemciler yanlış veri türlerini kullandığında sunucu bir hata döner.
- Örneğin, bazı verilerin Int olarak tanımlandığı bir yerde String kullanmak hatalara yol açar.
- Bu, geliştirme ekiplerinin API'nin tür doğrulamasını yapmasına olanak tanır.

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

- İstemcinin isteğindeki verileri içeren yanıtı oluşturmak için sunucu, çözümleyici işlevlerini kullanır.

- Çözümleyici işlevleri, istemcinin sorgusundaki her alan için yanıtı doldurma sorumluluğundadır.

- Bu yapmak için çözümleyiciler, ilişkisel veritabanlarına sorgu yapmak, önbellek veri tabanlarından veri okumak veya ağdaki diğer sunuculara HTTP istekleri yapmak gibi görevleri gerçekleştirebilirler.

BTK AKADEMİ

Her alanın yanıtını döndürmekten sorumlu bir çözümleyici (resolve) işlevi vardır.

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

- Çözümleyici işlevleri sadece veri tabanından okuma ile sınırlı değildir.
- Onlar yerel dosya sisteminden veri okuyabilir veya REST API'ler üzerinden ek sistemlere HTTP istekleri yapabilirler.
- Aslında, GraphQL API'leri genellikle özellikle şirketler REST'ten GraphQL'e geçiş yaparken, arka planda REST çağrıları yaparlar.

GraphQL Hangi Sorunları Çözer?

What Problems Does GraphQL Solve?

GraphQL Hangi Sorunları Çözer?

Aşırı veri çekimini (over-fetching) engeller

Yetersiz veri çekimini (under-fetching) engeller

Şema birleştirme

Şema federasyonu

Sorgu Ayrıştırıcı ve Çözümleyici İşlevleri

- Bazı durumlarda GraphQL, istemciler için görünmez olan çok sayıda arka uç REST hizmetini birleştirici API katmanı olarak kullanılır.

- GraphQL API ile etkileşimde bulunmak isteyen istemciler, Apollo Client gibi mevcut açık kaynak GraphQL istemci kütüphanelerini veya cURL gibi komut satırı HTTP istemcilerini kullanabilirler.

GraphQL vs REST API

http://lab.blackhatgraphql.com/start

Uygulamalar

- Sunucu Kurulumu
- GraphQL Kütüphanelerinin Yüklenmesi
- GraphQL Şema Tanımı
- Graphiql
- Sorgu Örnekleri

Neler Öğrendik?

1. Temeller
2. Kökeni
3. Kullanım Örnekleri
4. Spesifikasyon
5. GrapQL İletişimi
6. Şemalar
7. Sorgular
8. GraphQL Çözdüğü Sorunlar
9. GraphQL vs REST API
10. Uygulama Örneği