

# Resolving Queries

---



**Jonathan Mills**

CLOUD APPLICATION ARCHITECT

@jonathanfmills [www.jonathanfmills.com](http://www.jonathanfmills.com)



# Overview



Resolver maps

Executing queries

Schema directives

Implementing datasources

Query byId

Filtering queries

Code cleanup



Up Next:  
Implementing our resolver

---



# Implementing Resolvers

---



# Resolver

A function that is responsible for populating the data for a single field in your schema.



```
const typeDefs = gql`  
  type Query {  
    sessions:[Session]  
  }  
  type Session {  
    id: ID!  
    title: String!,  
    description:String,  
    startsAt:String,  
    endsAt:String,  
    room:String,  
    day:String,  
    format: String,  
    track:String,  
    level:String  
  }`
```

◀ Query Type

◀ Sessions Field



# Resolver Map

A big object that holds all of those  
 $\text{Type} \rightarrow \text{Field} \rightarrow \text{Resolver functions}.$



# Demo



Import dummy data

Implement our resolver





Up Next:  
Executing queries

---



# Executing Queries

---



Up Next:  
Field level directives

---



# Field Level Directives

---



# Field Level Directives

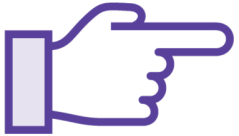
`@include(if: Boolean!)`

`@skip(if: Boolean!)`

`@deprecated(reason: String)`



# Field Level Directives



`@include(if: Boolean!)`

`@skip(if: Boolean!)`

`@deprecated(reason: String)`

**Calls the resolver for the annotated field if the boolean is true**



# Field Level Directives

`@include(if: Boolean!)`



`@skip(if: Boolean!)`

**The opposite of include.  
This will skip the field if the  
boolean is true.**

`@deprecated(reason: String)`



# Field Level Directives

`@include(if: Boolean!)`

`@skip(if: Boolean!)`

**Mark a field as deprecated with a reason for its future removal.**



`@deprecated(reason: String)`





# Up Next:

## Implementing Data Sources

---



# Implementing Data Sources

---



# Demo



## Implement a data source



Up Next:  
Filter By Id

---



# Demo



## Filter By Id



Up Next:  
Filtering By Many Columns

---



# Demo



## Filtering By Many Columns



Up Next:  
Cleaning Up Our Code

---





# Demo



## Cleaning Up Our Code



Up Next:  
Summary

---



# Summary



Resolver maps

Executing queries

Schema directives

Implementing datasources

Query byId

Filtering queries

Code cleanup



# Up Next:

## Adding Speakers to Sessions

---

