

Building Robust GraphQL APIs with AWS Amplify and AWS AppSync

Nader Dabit – AWS Amplify

August 19, 2019

What to expect from this session?

- Introduction to AWS Amplify
- Introduction to AWS AppSync
- Introduction to the GraphQL Transform library
- Walk through of GraphQL Transform directives
- Building a GraphQL API with GraphQL Transform Library
- Testing the API

What is AWS Amplify?



AWS Amplify is a CLI & toolchain for the client

What Does it Do?



CLI

Creates & configures AWS services

What Does it Do?



JavaScript Library

Connects your front-end application to cloud services

What kind of services?



- ✓ Database
- ✓ API
- ✓ Lambda / Serverless
- ✓ Authentication
- ✓ Analytics
- ✓ Hosting
- ✓ Storage
- ✓ ML / AI
- ✓ Chat Bots

Authentication



Amazon Cognito

- ✓ Create & configure new Amazon Cognito resources for user signup & signin
- ✓ Interact with Amazon Cognito using Auth class from client Library
- ✓ Pre-configured components available for React, React Native, Angular, & Ionic

GraphQL API



AWS AppSync

- ✓ Managed API service (GraphQL)
- ✓ Create new AWS AppSync GraphQL API
- ✓ Configure AppSync
- ✓ Perform queries, mutations, & register subscriptions from client applications
- ✓ GraphQL Transform library (Amplify CLI)

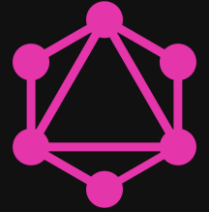
GraphQL API



GraphQL Transform

Provides an abstraction that helps you quickly create backends for your web and mobile applications on AWS.

GraphQL Transform



@model

@auth

@connection

@versioned

@key

@function

@searchable

GraphQL Transform



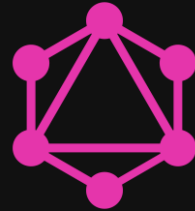
@model - top-level entity, creates DynamoDB table, resolvers & additional schema (queries, mutations & subscriptions) for base type

GraphQL Transform



```
21  type Todo {  
22    id: ID!  
23    name: String!  
24  }
```

GraphQL Transform



```
21  type Todo @model {  
22    id: ID!  
23    name: String!  
24  }
```

GraphQL Transform

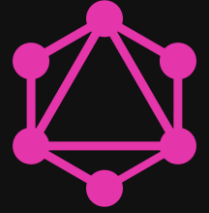


```
21   type Todo @model {  
22     id: ID!  
23     name: String!  
24   }
```

@model

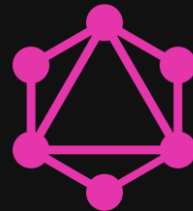
- + CRUD & List operations
- + DynamoDB Table
- + Subscriptions
- + Resolvers for all operations

GraphQL Transform



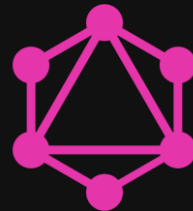
@connection - enables relationships between @model types

GraphQL Transform



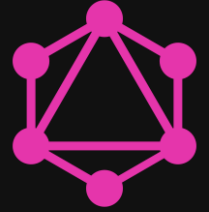
```
4  type Post @model {
5    id: ID!
6    title: String!
7    comments: [Comment]
8  }
9
10 type Comment @model {
11   id: ID!
12   content: String!
13   createdAt: String
14 }
```


GraphQL Transform



```
4  type Post @model {
5    id: ID!
6    title: String!
7    comments: [Comment] @connection(name: "PostComments", sortField: "createdAt")
8  }
9
10 type Comment @model {
11   id: ID!
12   content: String!
13   post: Post @connection(name: "PostComments", sortField: "createdAt")
14   createdAt: String
15 }
```

GraphQL Transform

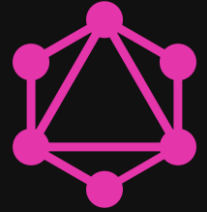


```
4  type Post @model {
5    id: ID!
6    title: String!
7    comments: [Comment] @connection(name: "PostComments", sortField: "createdAt")
8  }
```

@connection

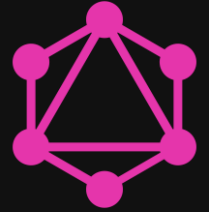
Model relationships between data. CLI creates the necessary resolvers.

GraphQL Transform



@auth - enables ability to specify authentication rules

GraphQL Transform



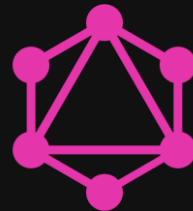
```
5  type Post @model
6      id: ID!
7      title: String!
8  }
```

GraphQL Transform



```
4  type Post @model @auth(rules: [{ allow: owner, queries: null }]) {  
5    id: ID!  
6    title: String!  
7  }
```

GraphQL Transform



```
4   type Post @model @auth(rules: [{ allow: owner, queries: null }]) {  
5     id: ID!  
6     title: String!  
7   }
```

@auth

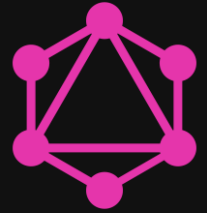
Define authorization rules

GraphQL Transform



@key – enables ability to configure custom index structures for @model types

GraphQL Transform



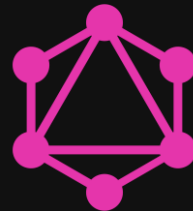
```
24  type Event @model {  
25    id: ID!  
26    name: String!  
27    createdAt: String!  
28  }
```


GraphQL Transform



```
22 type Event @model @key(name: "queryName", fields: ["queryName", "createdAt"], queryField: "itemsByDate") {  
23   id: ID!  
24   name: String!  
25   createdAt: String!  
26   queryName: String!  
27 }
```

GraphQL Transform



```
22 type Event @model @key(name: "queryName", fields: ["queryName", "createdAt"], queryField: "itemsByDate") {  
23   id: ID!  
24   name: String!  
25   createdAt: String!  
26   queryName: String!  
27 }
```

@key

Configure custom index structures

AWS Amplify CLI

Getting started

Installing the CLI



```
npm install -g @aws-amplify/cli
```

AWS Amplify CLI

Getting started

Initializing a new project



```
amplify init
```

AWS Amplify CLI

Getting started

Adding a feature



```
amplify add auth
```



DEMO