

API GATEWAY

GRAPHQL TESTING

REPOSITORY

- ▶ <https://github.com/aofleejay/graphql-testing-workshop>
- ▶ Starter branch - workshop/checkpoint-0
- ▶ `npm install`

OUTLINE

- ▶ Unit testing
 - ▶ Schema testing
 - ▶ Resolver testing
- ▶ Integration testing

UNIT TESTING

SCHEMA TESTING

SCHEMA TESTING

► src/schemas/characters.js

```
const typeDefs = `
  type Character {
    id: String!
    name: String
    gender: Gender
  }

  type Gender {
    name: String
  }
`
```

SCHEMA TESTING

► src/schemas/rootSchema.js

```
import { makeExecutableSchema } from 'graphql-tools'

const schema = makeExecutableSchema({
  typeDefs,
  resolvers: charactersResolvers,
})
```

SCHEMA TESTING

► src/schemas/rootSchema.js

```
import { makeExecutableSchema } from 'graphql-tools'
```

```
const schema = makeExecutableSchema({  
  typeDefs,  
  resolvers: charactersResolvers,  
})
```

FIND SCHEMA

► `console.log(schema)`

```
GraphQLSchema {  
  ...  
  _typeMap:  
    {  
      Query: Query,  
      Character: Character,  
      String: String,  
      Gender: Gender,  
    }  
}
```


GET FIELDS

► Character.getFields()

```
{
  id: {
    type: String!,
    ...
  }
  name: {
    type: String,
    ...
  }
}
```

GRAPHQL TYPES

```
{  
  id: {  
    type: String!,  
    ...  
  }  
  name: {  
    type: String,  
    ...  
  }  
}
```

```
import {  
  GraphQLString,  
  GraphQLBoolean,  
  GraphQLInt  
} from 'graphql'
```

SCHEMA TESTING

▶ test/unit/characters.spec.js

```
import { GraphQLString } from 'graphql'
```

```
const characterSchema = rootSchema._typeMap.Character.getFields()
```

```
expect(characterSchema).to.have.property('name')
```

```
expect(characterSchema.name.type).to.deep.equal(GraphQLString)
```

CHECKPOINT 1

- ▶ [workshop/checkpoint-1](#)

UNIT TESTING

RESOLVER TESTING

RESOLVER TESTING

► src/schemas/characters.js

```
GraphQLSchema {  
  ...  
  _typeMap:  
  {  
    Query: Query,  
    Character: Character,  
    String: String,  
    Gender: Gender,  
  }  
}
```

GET FIELDS

► `Query.getFields()`

```
{
  characters: {
    type: [Character],
    ...
    resolve: [Function: characters]
  }
}
```

RESOLVER TESTING

► test/unit/characters.spec.js

```
const charactersQuery = rootSchema._typeMap.Query.getFields().characters
```

```
charactersQuery.resolve()  
  .then(res => {  
    expect(res).to.deep.equals(expected)  
  })
```


CHECKPOINT 2

- ▶ [workshop/checkpoint-2](#)

INTEGRATION TESTING

SUPERTEST

SUPERTEST

- ▶ <https://www.npmjs.com/package/supertest>

SUPERTEST

```
request.agent(server)
  .post('/graphql')
  .set('Accept', 'application/json')
  .send(query)
  .then(res => {
    console.log(res.body)
  })
```

CHECKPOINT 3

- ▶ [workshop/checkpoint-3](#)

FEEDBACK PLS

- ▶ <https://docs.google.com/forms/d/1FuE1nVsbwcHcrVZdSaCaI7BFryeEfa1I1vXD9J65tM>