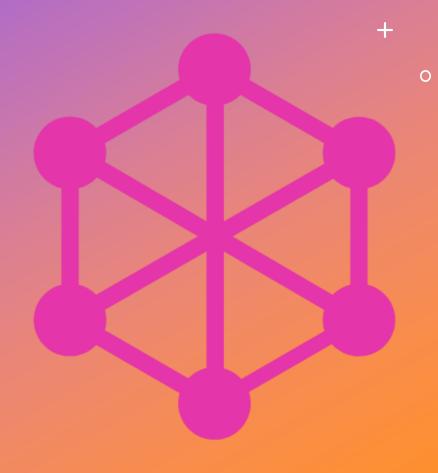
## UPDATING THE RESOLVE FUNCTIONS

in this GraphQL tutorial we'll go through all of our resolve functions and update them, so that now they interact with our MongoDB instance, instead of local dummy data.



```
const graphql = require('graphql');
const Book = require('../models/book');
const Author = require('../models/Author');
const _ = require('lodash');
const {
   GraphQLObjectType,
    GraphQLString,
    GraphQLSchema,
   GraphQLID,
    GraphQLInt,
    GraphQLList
} = graphql;
const BookType = new GraphQLObjectType({
    name: 'Book',
    fields: ( ) => ({
        id: { type: GraphQLID },
                                                                              Modify
        name: { type: GraphQLString },
                                                                              code
        genre: { type: GraphQLString },
        author: {
            type: AuthorType,
            resolve(parent, args){
               return Author.findById(parent.authorId);
```

});

shema.js

```
const graphql = require('graphql');
const Book = require('../models/book');
const Author = require('../models/Author');
const _ = require('lodash');
const {
   GraphQLObjectType,
   GraphQLString,
   GraphQLSchema,
   GraphQLID,
   GraphQLInt,
   GraphQLList
} = graphql;
const BookType = new GraphQLObjectType({...});
const AuthorType = new GraphQLObjectType({
   name: 'Author',
                                                                              Modify
   fields: ( ) => ({
                                                                              code
        id: { type: GraphQLID },
        name: { type: GraphQLString },
        age: { type: GraphQLInt },
        books: {
           type: new GraphQLList(BookType),
           resplve(parent, args){
               return Book.find({ authorId: parent.id });
    })
});
```

shema.js

shema.js

. . .

```
const RootQuery = new GraphQLObjectType({
    name: 'RootQueryType',
    fields: {
        book: {
            type: BookType,
            args: { id: { type: GraphQLID } },
            resolve(parent, args){
                return Book.findById(args.id);
        },
        author: {
            type: AuthorType,
            args: { id: { type: GraphQLID } },
            resolve(parent, args){
               return Author.findById(args.id);
        },
        books: {
            type: new GraphQLList(BookType),
                                                                                       Modify
            resolve(parent, args){
                                                                                       code
                return Book.find({});
        },
        authors: {
            type: new GraphQLList(AuthorType),
            resolve(parent, args){
                return Author.find({});
});
```