const graphql = require("graphql");

var \_ = require("lodash");

var userData = [

  { id: "1", name: "Samir", age: 31 },

  {

    id: "2",

    name: "Mounica",

    age: 26

  }

];

var hobbiesData = [

  { id: "1", title: "Programmin", description: "EnjoyingProgramming" },

  { id: "2", title: "Movies", description: "WatchingMovies" }

];

var postData = [

  { id: "1", comment: "Your are awesome" },

  { id: "2", comment: "you are intelligent" }

];

const {

  GraphQLObjectType,

  GraphQLID,

  GraphQLString,

  GraphQLInt,

  GraphQLSchema

} = graphql;

const UserType = new GraphQLObjectType({

  name: "User",

  description: "Use Schema Info..",

  fields: () => ({

    id: { type: GraphQLString },

    name: { type: GraphQLString },

    age: { type: GraphQLInt }

  })

});

const UserHobbyType = new GraphQLObjectType({

  name: "Hobbies",

  description: "hobby description...",

  fields: () => ({

    id: { type: GraphQLString },

    title: { type: GraphQLString },

    description: { type: GraphQLString }

  })

});

const postType = new GraphQLObjectType({

  name: "UserPostType",

  description: "Posted Comments of User",

  fields: () => ({

    id: { type: GraphQLString },

    comment: { type: GraphQLString }

  })

});

const RootQuery = new GraphQLObjectType({

  name: "RootQueryType",

  description: "Description",

  fields: {

    user: {

      type: UserType,

      args: { id: { type: GraphQLString } },

      resolve(parent, args) {

        return \_.find(userData, { id: args.id });

        //We Resolve the data Here

        //Get the Data from the Data Source

      }

    },

    hobby: {

      type: UserHobbyType,

      args: { id: { type: GraphQLString } },

      resolve(parent, args) {

        return \_.find(hobbiesData, { id: args.id });

      }

    },

    userPost: {

      type: postType,

      args: { id: { type: GraphQLString } },

      resolve(parent, args) {

        return \_.find(postData, { id: args.id });

      }

    }

  }

});

module.exports = new GraphQLSchema({

  query: RootQuery

});