```
#! /usr/bin/env node

console.log('This script populates some test books, authors, genres and bookinstances to your
database. Specified database as argument - e.g.: populatedb
mongodb+srv://cooluser:coolpassword@cluster0.a9azn.mongodb.net/local_library?retryWrites=true');

// Get arguments passed on command line
var userArgs = process.argv.slice(2);
/*
if (!userArgs[0].startsWith('mongodb')) {
    console.log('ERROR: You need to specify a valid mongodb URL as the first argument');
    return
}
*/
var async = require('async')
var Book = require('./models/book')
var Author = require('./models/author')
var Genre = require('./models/genre')
var BookInstance = require('./models/bookinstance')


var mongoose = require('mongoose');
var mongoDB = userArgs[0];
mongoose.connect(mongoDB, {useNewUrlParser: true, useUnifiedTopology: true});
mongoose.Promise = global.Promise;
var db = mongoose.connection;
db.on('error', console.error.bind(console, 'MongoDB connection error:'));

var authors = []
var genres = []
var books = []
var bookinstances = []

function authorCreate(first_name, family_name, d_birth, d_death, cb) {
  authordetail = {first_name:first_name , family_name: family_name }
  if (d_birth != false) authordetail.date_of_birth = d_birth
  if (d_death != false) authordetail.date_of_death = d_death

  var author = new Author(authordetail);

  author.save(function (err) {
    if (err) {
      cb(err, null)
      return
    }
    console.log('New Author: ' + author);
    authors.push(author)
    cb(null, author)
  }  );
}

function genreCreate(name, cb) {
  var genre = new Genre({ name: name });

  genre.save(function (err) {
    if (err) {
      cb(err, null);
      return;
    }
    console.log('New Genre: ' + genre);
    genres.push(genre)
    cb(null, genre);
  }   );
}

function bookCreate(title, summary, isbn, author, genre, cb) {
  bookdetail = {
    title: title,
```

```
    summary: summary,
    author: author,
    isbn: isbn
  }
  if (genre != false) bookdetail.genre = genre

  var book = new Book(bookdetail);
  book.save(function (err) {
    if (err) {
      cb(err, null)
      return
    }
    console.log('New Book: ' + book);
    books.push(book)
    cb(null, book)
  }  );
}


function bookInstanceCreate(book, imprint, due_back, status, cb) {
  bookinstancedetail = {
    book: book,
    imprint: imprint
  }
  if (due_back != false) bookinstancedetail.due_back = due_back
  if (status != false) bookinstancedetail.status = status

  var bookinstance = new BookInstance(bookinstancedetail);
  bookinstance.save(function (err) {
    if (err) {
      console.log('ERROR CREATING BookInstance: ' + bookinstance);
      cb(err, null)
      return
    }
    console.log('New BookInstance: ' + bookinstance);
    bookinstances.push(bookinstance)
    cb(null, book)
  }  );
}


function createGenreAuthors(cb) {
    async.series([
        function(callback) {
          authorCreate('Patrick', 'Rothfuss', '1973-06-06', false, callback);
        },
        function(callback) {
          authorCreate('Ben', 'Bova', '1932-11-8', false, callback);
        },
        function(callback) {
          authorCreate('Isaac', 'Asimov', '1920-01-02', '1992-04-06', callback);
        },
        function(callback) {
          authorCreate('Bob', 'Billings', false, false, callback);
        },
        function(callback) {
          authorCreate('Jim', 'Jones', '1971-12-16', false, callback);
        },
        function(callback) {
          genreCreate("Fantasy", callback);
        },
        function(callback) {
          genreCreate("Science Fiction", callback);
        },
        function(callback) {
          genreCreate("French Poetry", callback);
        },
        ],
        // optional callback
```

```
        cb);
}


function createBooks(cb) {
    async.parallel([
        function(callback) {
          bookCreate('The Name of the Wind (The Kingkiller Chronicle, #1)', 'I have stolen princesses back from sleeping barrow kings. I burned down the town of Trebon. I have spent the night with Felurian and left with both my sanity and my life. I was expelled from the University at a younger age than most people are allowed in. I tread paths by moonlight that others fear to speak of during day. I have talked to Gods, loved women, and written songs that make the minstrels weep.', '9781473211896', authors[0], [genres[0],], callback);
        },
        function(callback) {
          bookCreate("The Wise Man's Fear (The Kingkiller Chronicle, #2)", 'Picking up the tale of Kvothe Kingkiller once again, we follow him into exile, into political intrigue, courtship, adventure, love and magic... and further along the path that has turned Kvothe, the mightiest magician of his age, a legend in his own time, into Kote, the unassuming pub landlord.', '9788401352836', authors[0], [genres[0],], callback);
        },
        function(callback) {
          bookCreate("The Slow Regard of Silent Things (Kingkiller Chronicle)", 'Deep below the University, there is a dark place. Few people know of it: a broken web of ancient passageways and abandoned rooms. A young woman lives there, tucked among the sprawling tunnels of the Underthing, snug in the heart of this forgotten place.', '9780756411336', authors[0], [genres[0],], callback);
        },
        function(callback) {
          bookCreate("Apes and Angels", "Humankind headed out to the stars not for conquest, nor exploration, nor even for curiosity. Humans went to the stars in a desperate crusade to save intelligent life wherever they found it. A wave of death is spreading through the Milky Way galaxy, an expanding sphere of lethal gamma ...", '9780765379528', authors[1], [genres[1],], callback);
        },
        function(callback) {
          bookCreate("Death Wave","In Ben Bova's previous novel New Earth, Jordan Kell led the first human mission beyond the solar system. They discovered the ruins of an ancient alien civilization. But one alien AI survived, and it revealed to Jordan Kell that an explosion in the black hole at the heart of the Milky Way galaxy has created a wave of deadly radiation, expanding out from the core toward Earth. Unless the human race acts to save itself, all life on Earth will be wiped out...", '9780765379504', authors[1], [genres[1],], callback);
        },
        function(callback) {
          bookCreate('Test Book 1', 'Summary of test book 1', 'ISBN111111', authors[4], [genres[0],genres[1]], callback);
        },
        function(callback) {
          bookCreate('Test Book 2', 'Summary of test book 2', 'ISBN222222', authors[4], false, callback)
        }
        ],
        // optional callback
        cb);
}


function createBookInstances(cb) {
    async.parallel([
        function(callback) {
          bookInstanceCreate(books[0], 'London Gollancz, 2014.', false, 'Available', callback)
        },
        function(callback) {
          bookInstanceCreate(books[1], ' Gollancz, 2011.', false, 'Loaned', callback)
        },
        function(callback) {
          bookInstanceCreate(books[2], ' Gollancz, 2015.', false, false, callback)
        },
        function(callback) {
          bookInstanceCreate(books[3], 'New York Tom Doherty Associates, 2016.', false,
```

```
  'Available', callback)
          },
          function(callback) {
            bookInstanceCreate(books[3], 'New York Tom Doherty Associates, 2016.', false,
  'Available', callback)
          },
          function(callback) {
            bookInstanceCreate(books[3], 'New York Tom Doherty Associates, 2016.', false,
  'Available', callback)
          },
          function(callback) {
            bookInstanceCreate(books[4], 'New York, NY Tom Doherty Associates, LLC, 2015.', false,
  'Available', callback)
          },
          function(callback) {
            bookInstanceCreate(books[4], 'New York, NY Tom Doherty Associates, LLC, 2015.', false,
  'Maintenance', callback)
          },
          function(callback) {
            bookInstanceCreate(books[4], 'New York, NY Tom Doherty Associates, LLC, 2015.', false,
  'Loaned', callback)
          },
          function(callback) {
            bookInstanceCreate(books[0], 'Imprint XXX2', false, false, callback)
          },
          function(callback) {
            bookInstanceCreate(books[1], 'Imprint XXX3', false, false, callback)
          }
          ],
          // Optional callback
          cb);
  }



  async.series([
      createGenreAuthors,
      createBooks,
      createBookInstances
  ],
  // Optional callback
  function(err, results) {
      if (err) {
          console.log('FINAL ERR: '+err);
      }
      else {
          console.log('BOOKInstances: '+bookinstances);

      }
      // All done, disconnect from database
      mongoose.connection.close();
  });
```