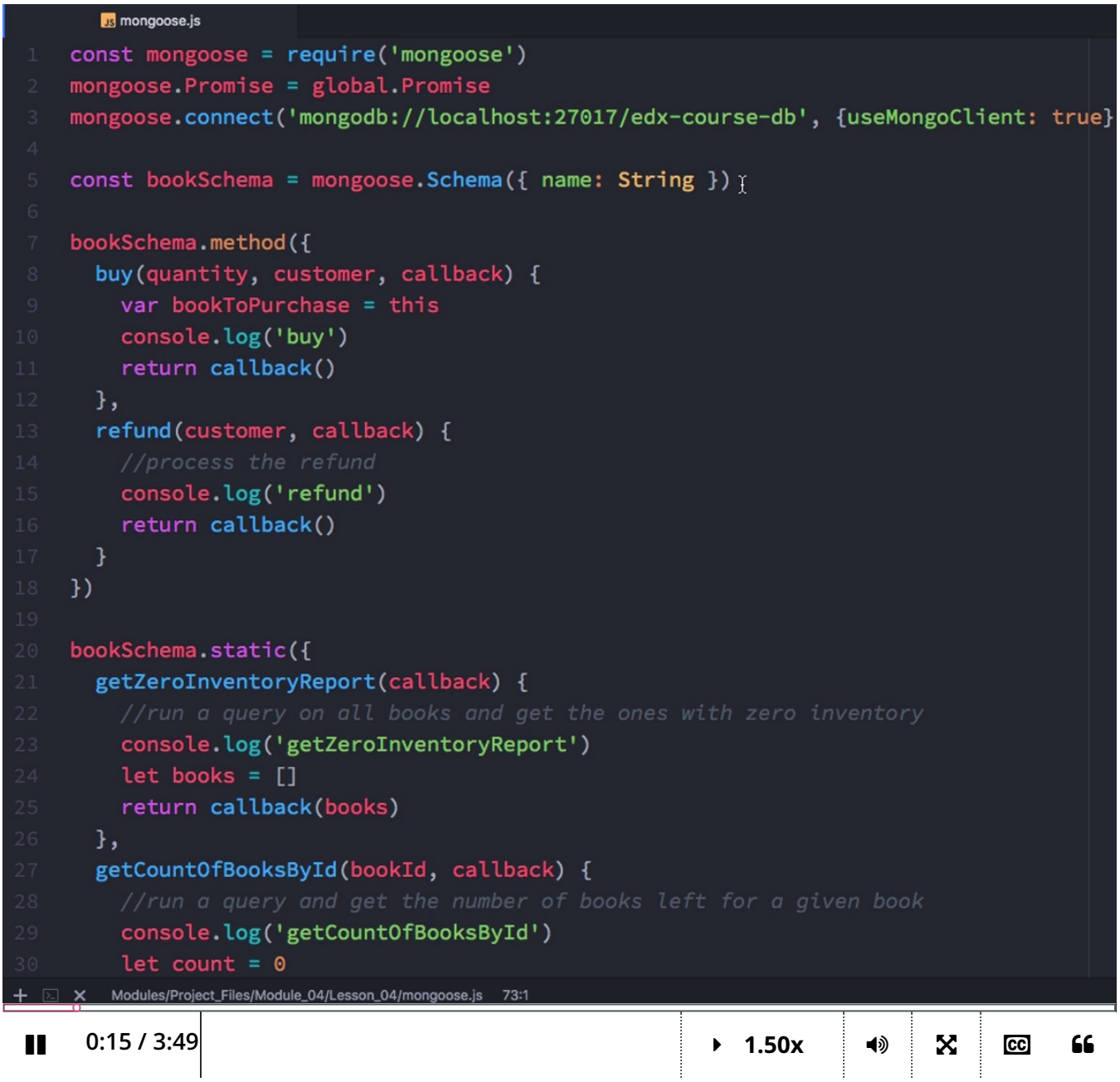
Custom Mongoose Methods
Video: Custom Mongoose Methods



Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Custom Static and Instance Methods

In addition to dozens of built-in Mongoose model methods, we can add custom ones. For example, to initiate a purchase, we can call the `buy` method on the document `practicalNodeBook` after we implement the custom instance method `buy()`:

```
bookSchema.method({
  buy: function(quantity, customer, callback) {
    var bookToPurchase = this
    //create a purchase order and invoice customer
    return callback(results)
  },
  refund: function(customer, callback) {
    //process the refund
    return callback(results)
  }
})
```

Static methods are useful when we either don't have a particular document object or we don't need it:

```
bookSchema.static({
  getZeroInventoryReport: function(callback) {
    //run a query on all books and get the ones with zero inventory
    return callback(books)
  },
  getCountOfBooksById: function(bookId, callback){
    //run a query and get the number of books left for a given book
    return callback(count)
  }
})
```

Hooks for Keeping Code Organized

In a complex application with a lot of interrelated objects, we might want to execute certain logic before saving an object. Hooks are a good place to store such logic. For example, we might want to upload a PDF to the web site before saving a book document:

```
bookSchema.pre('save', function(next) {
    //prepare for saving
    //upload PDF
    return next()
})
```

On the other hand, before removing, we need to make sure there are no pending purchase orders for this book:

```
bookSchema.pre('remove', function(next) {
    //prepare for removing
    return next(e)
})
```

Hooks are triggered before or after an event is executed. Hooks allow you to place business logic where it's best suited - in the model schemas.

Note: Hooks and methods must be added to the schemas before compiling them to models—in other words, before calling the `mongoose.model()` method.