

INLS 672

Web Development 2

Node.js Application



UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

Part 1: Overview, Installation

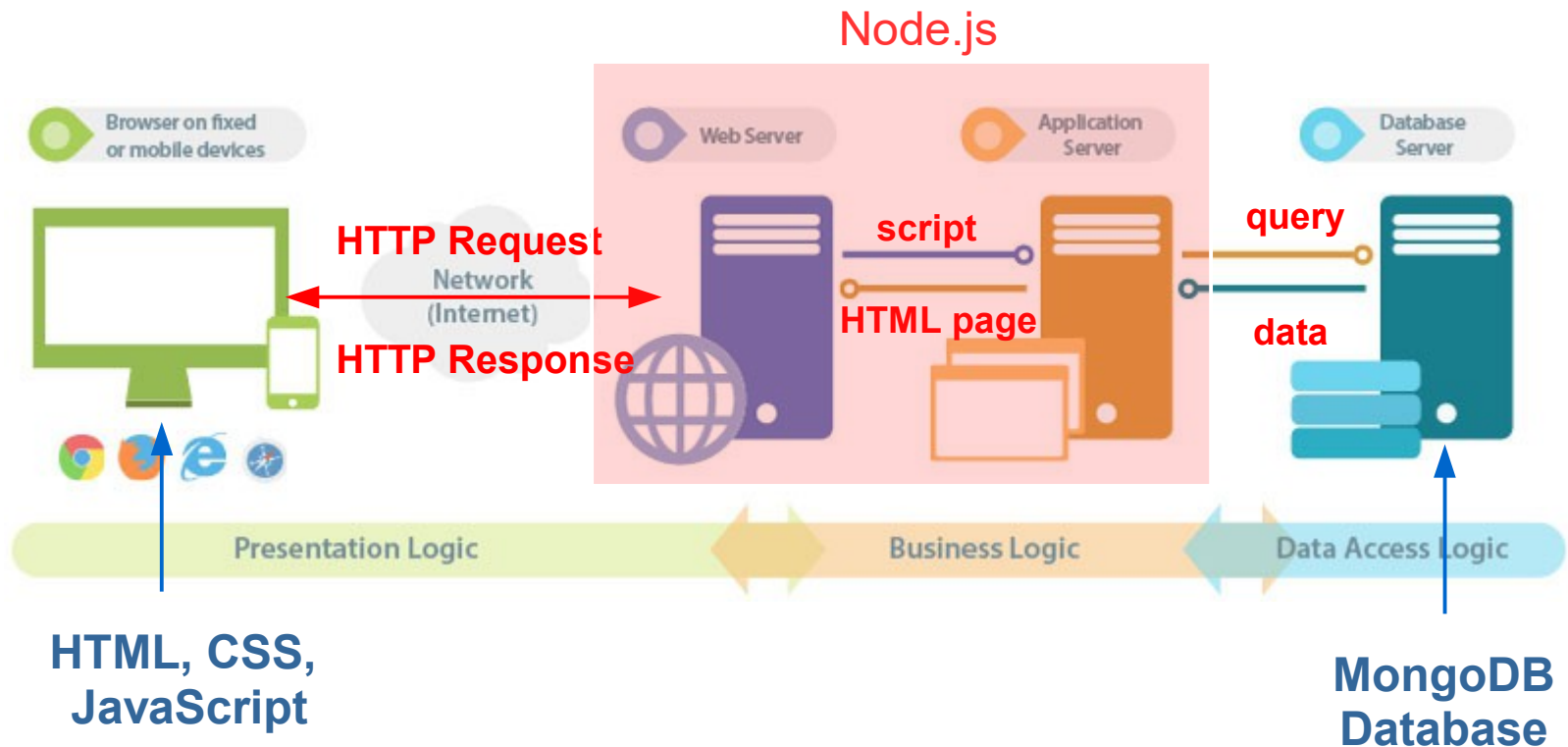
Part 2: Views

Part 3: Models

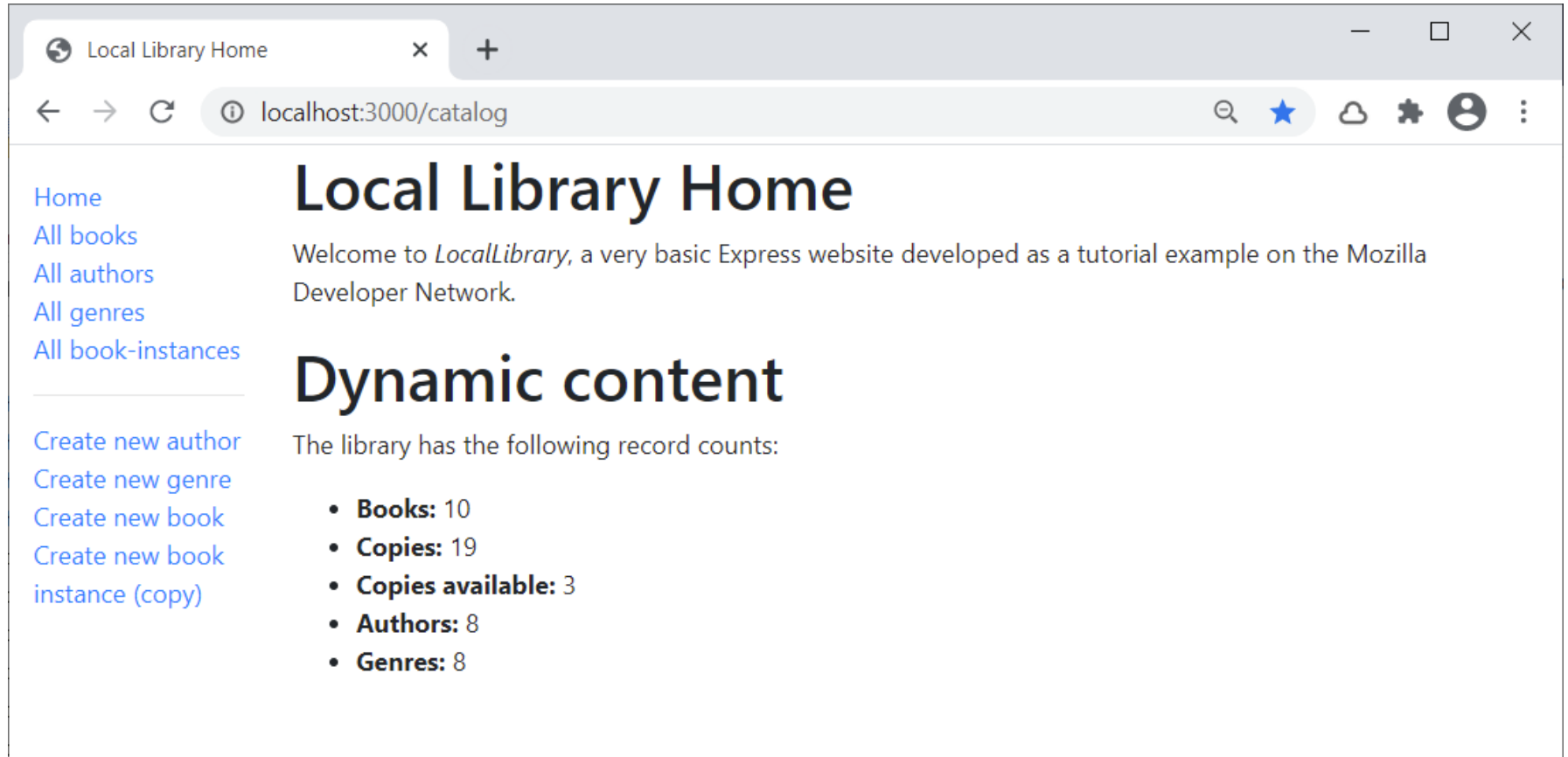
Part 4: Routes and Controllers

Node.js Web Application

The [Local Library web application](#) is built on the Node.js server and uses a MongoDB database to store information about books, authors, book instances, and genres.



Home Page for Local Library Website



A Node.js Web Application

The [Local Library web application](#) is a tutorial at MDN Web Docs.

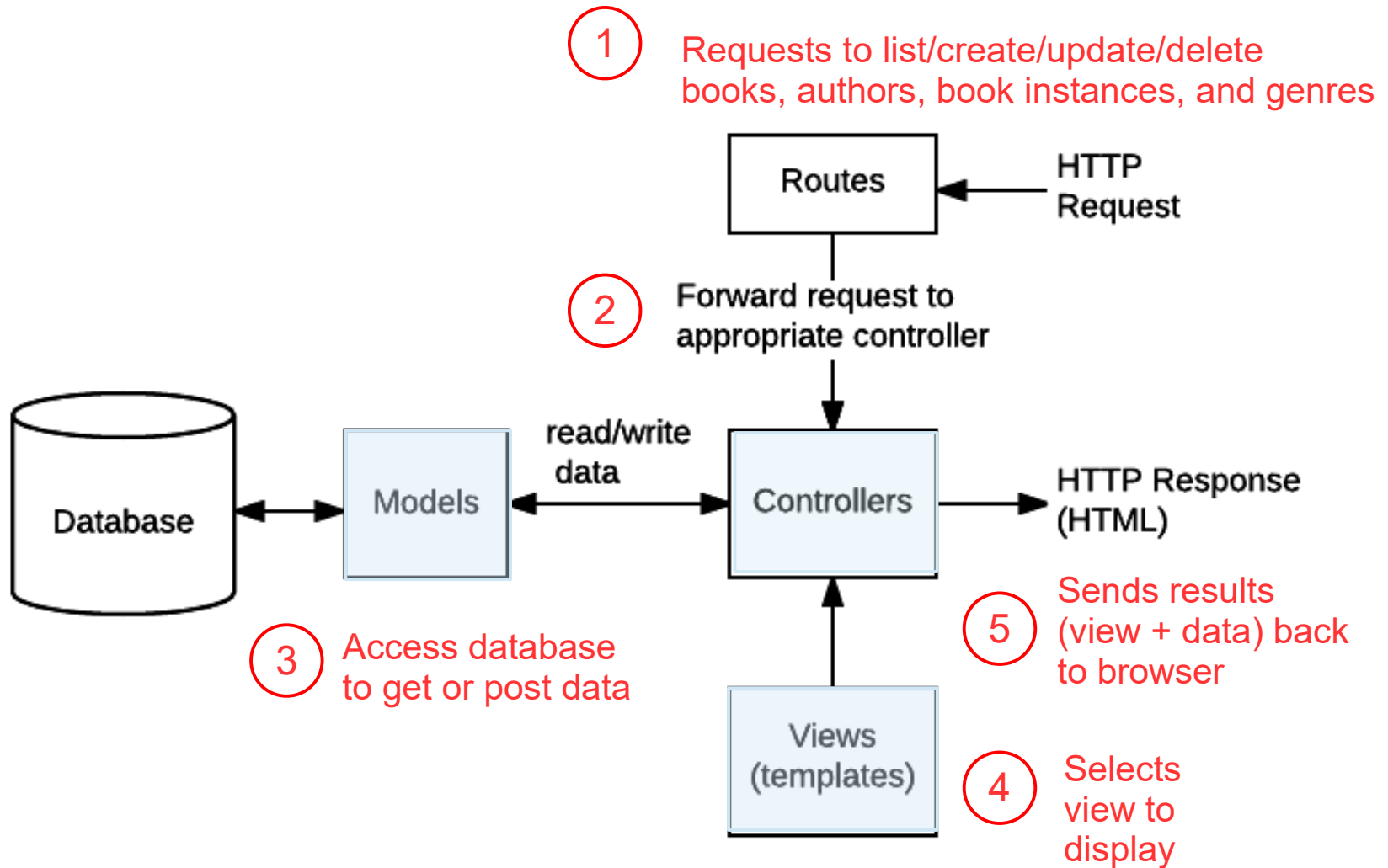
- The design is based on the MVC pattern, and the tutorial takes you through the steps to create the Local Library website.
- Uses the Express App Generator tool to create a skeleton website
- What the application does...
 - Creates and starts a Node.js web server
 - Supports the Create, Read, Update, and Delete (CRUD) operations
 - Defines web forms for collecting user input (**View**)
 - Uses a MongoDB database to store data (**Model**)
 - Defines **Controllers** for requesting different information, and templates (**Views**) to render the data as HTML in the browser

The complete application is available on the course website.


IMPORTANT: you are NOT required to work through the tutorial for this class, nor are you responsible for the (very detailed) content in the tutorial.

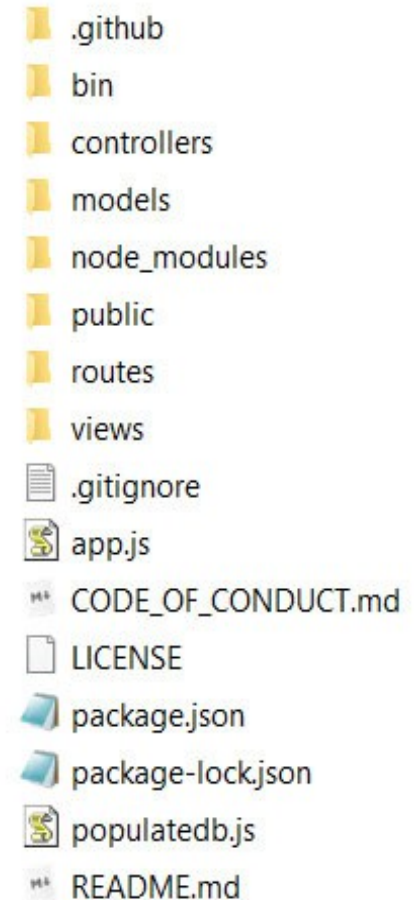
However, if you want to learn more about Node.js, and have a strong programming background, then this is a good learning tool.

Node.js apps use the MVC pattern



Installing the Local Library app

- Download `express-locallibrary-tutorial-master.zip` from the Examples folder and extract to your `NodeProjects` folder
- This creates the `express-locallibrary-tutorial-master` folder with all of the files needed to run the Local Library app
- These sub-folders should appear in your folder 



Running the Local Library app (Windows)

- 1) Navigate to the new folder, [express-locallibrary-tutorial-master](#)
- 2) Enter at the command prompt: `npm start` You should see output similar to the screenshot below on Windows:

```
npm
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\joanp>cd NodeProjects

C:\Users\joanp\NodeProjects>cd express-locallibrary-tutorial-master

C:\Users\joanp\NodeProjects\express-locallibrary-tutorial-master>npm start

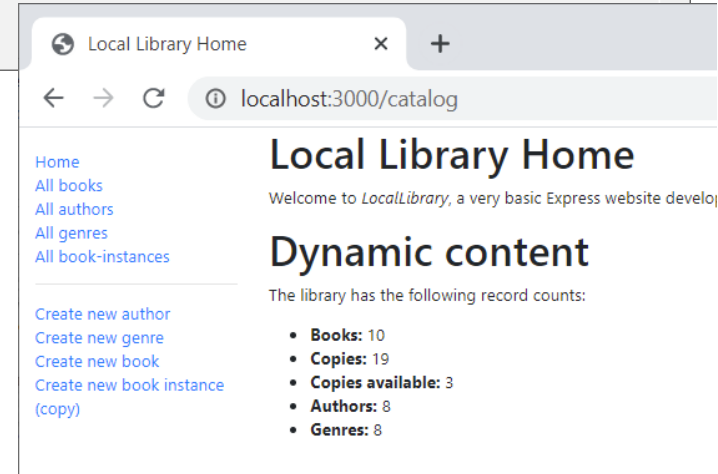
> express-locallibrary-tutorial@0.0.0 start C:\Users\joanp\NodeProjects\express-locallibrary-tutorial-master
> node ./bin/www

(node:13884) Warning: Accessing non-existent property 'MongoError' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:13884) DeprecationWarning: Listening to events on the Db class has been deprecated and will be removed in the next major version.
GET /catalog 304 512.353 ms - -
GET /stylesheets/style.css 304 1.184 ms - -
```

HTTP requests for the HTML and CSS

- 3) Enter in the browser: localhost:3000

Note: the 2 warnings are likely related to issues in the Node modules library, but do not seem to affect the application



Running the Local Library app (Mac)

- 1) Navigate to the new folder, `express-locallibrary-tutorial-master`
- 2) Enter at the command prompt: `npm start` You should see output similar to the screenshot below on Mac:

```
express-locallibrary-tutorial-master — node • npm TERM_PROGRAM=Apple_Terminal SHELL=/bin/bash — 155x24
Last login: Fri Mar 5 15:57:26 on console
[Joans-MacBook-Air:~ joan$ cd NodeProjects
[Joans-MacBook-Air:NodeProjects joan$ cd express-locallibrary-tutorial-master
[Joans-MacBook-Air:express-locallibrary-tutorial-master joan$ npm start

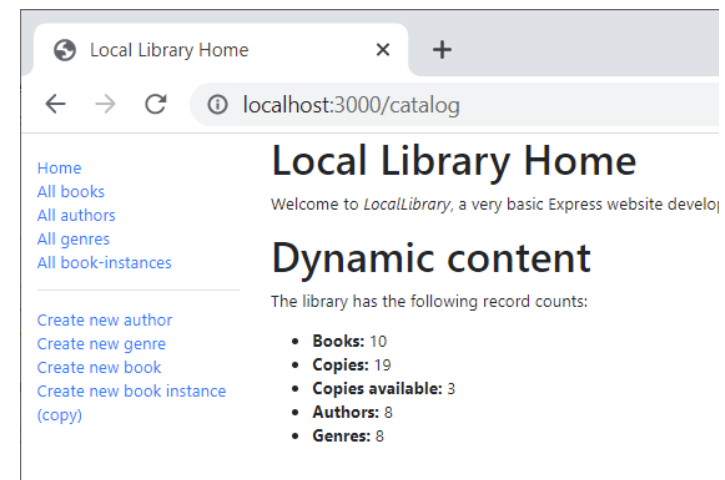
> express-locallibrary-tutorial@0.0.0 start /Users/joan/NodeProjects/express-locallibrary-tutorial-master
> node ./bin/www

(node:1748) Warning: Accessing non-existent property 'MongoError' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:1748) DeprecationWarning: Listening to events on the Db class has been deprecated and will be removed in the next major version.
GET / 302 11.533 ms - 60
GET /catalog 200 614.058 ms - -
GET /stylesheets/style.css 200 5.960 ms - 76
GET /favicon.ico 404 47.051 ms - -
```

HTTP requests for the HTML and CSS

- 3) Enter in the browser: `localhost:3000`

Note: the 2 warnings are likely related to issues in the Node modules library, but do not seem to effect the application

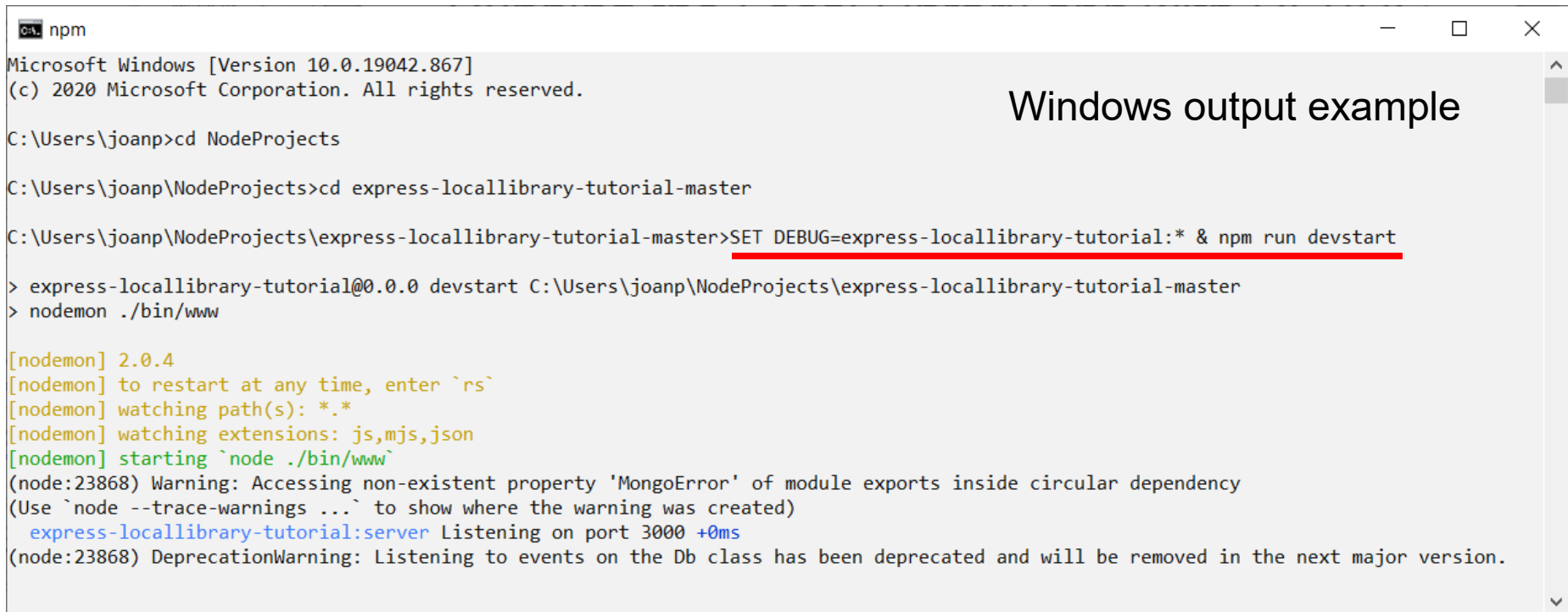


Running the Local Library app with DEBUG

You can run the application in DEBUG mode and additional information will be provided. Instead of entering `npm start`, enter:

Windows: `SET DEBUG=express-locallibrary-tutorial:* & npm run devstart`

Mac: `DEBUG=express-locallibrary-tutorial:* npm run devstart`



```
npm
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\joanp>cd NodeProjects

C:\Users\joanp\NodeProjects>cd express-locallibrary-tutorial-master

C:\Users\joanp\NodeProjects\express-locallibrary-tutorial-master>SET DEBUG=express-locallibrary-tutorial:* & npm run devstart

> express-locallibrary-tutorial@0.0.0 devstart C:\Users\joanp\NodeProjects\express-locallibrary-tutorial-master
> nodemon ./bin/www

[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./bin/www`
(node:23868) Warning: Accessing non-existent property 'MongoError' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
  express-locallibrary-tutorial:server Listening on port 3000 +0ms
(node:23868) DeprecationWarning: Listening to events on the Db class has been deprecated and will be removed in the next major version.
```

Windows output example

Part 1: Overview, Installation

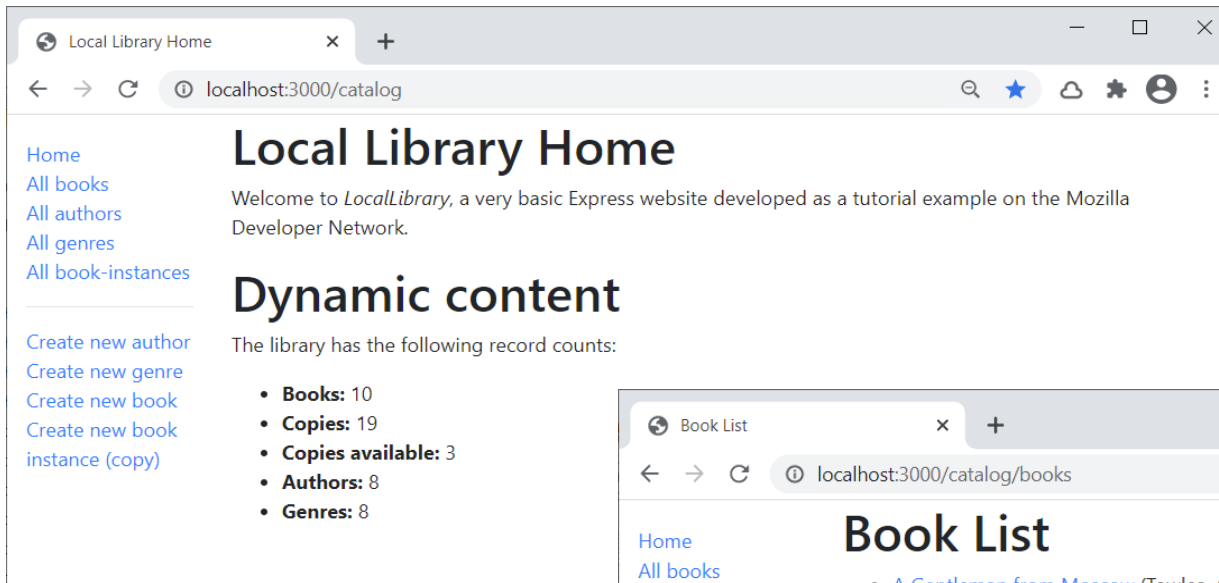
Part 2: Views

Part 3: Models

Part 4: Routes and Controllers

Local Library Views

- Four models: *Book*, *Author*, *Book Instance*, *Genre*
- For each model, there are 4 views: *List*, *Form*, *Detail*, *Delete*



The screenshot shows a web browser window with the title "Local Library Home" and the URL "localhost:3000/catalog". The page features a navigation menu on the left with links for Home, All books, All authors, All genres, All book-instances, and Create new author/book/instance. The main content area has a heading "Local Library Home" followed by a welcome message. Below that is a section titled "Dynamic content" with a list of record counts: Books (10), Copies (19), Copies available (3), Authors (8), and Genres (8).

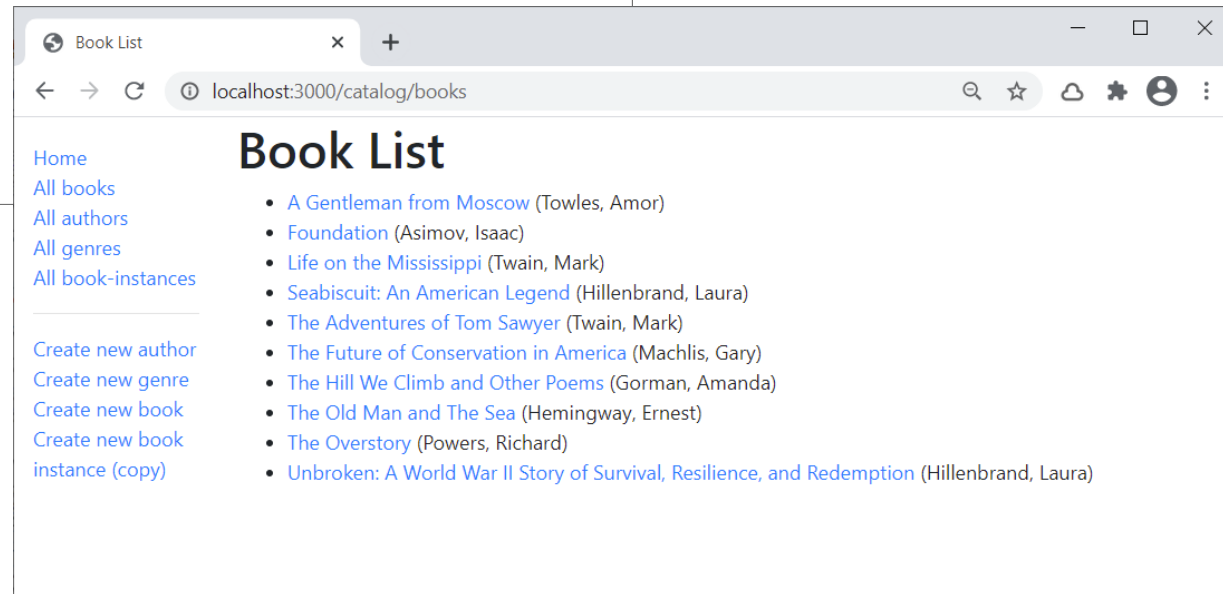
Local Library Home

Welcome to *LocalLibrary*, a very basic Express website developed as a tutorial example on the Mozilla Developer Network.

Dynamic content

The library has the following record counts:

- **Books:** 10
- **Copies:** 19
- **Copies available:** 3
- **Authors:** 8
- **Genres:** 8



The screenshot shows a web browser window with the title "Book List" and the URL "localhost:3000/catalog/books". The page features a navigation menu on the left with links for Home, All books, All authors, All genres, All book-instances, and Create new author/book/instance. The main content area has a heading "Book List" followed by a list of book titles and authors.

Book List

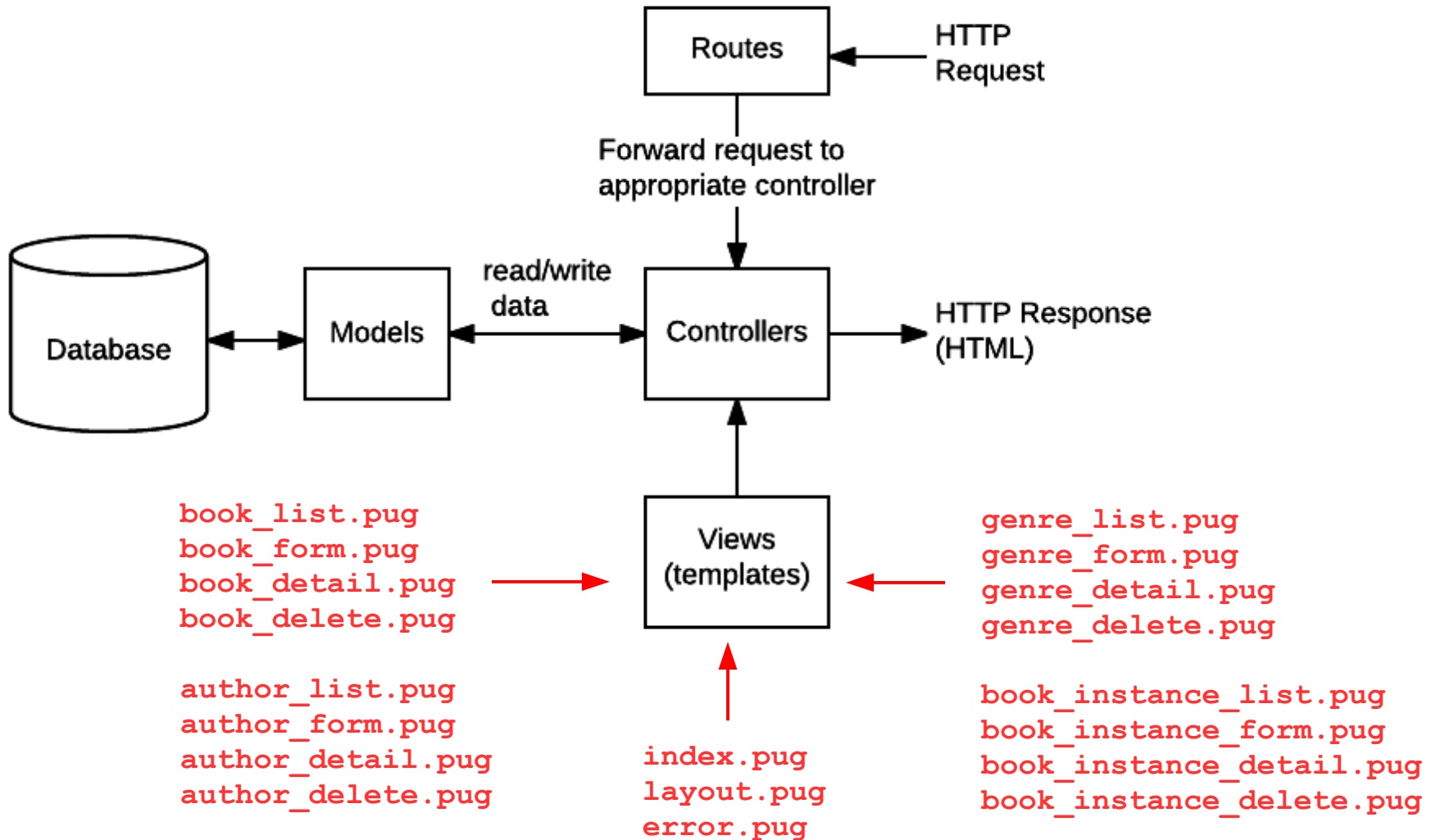
- [A Gentleman from Moscow](#) (Towles, Amor)
- [Foundation](#) (Asimov, Isaac)
- [Life on the Mississippi](#) (Twain, Mark)
- [Seabiscuit: An American Legend](#) (Hillenbrand, Laura)
- [The Adventures of Tom Sawyer](#) (Twain, Mark)
- [The Future of Conservation in America](#) (Machlis, Gary)
- [The Hill We Climb and Other Poems](#) (Gorman, Amanda)
- [The Old Man and The Sea](#) (Hemingway, Ernest)
- [The Overstory](#) (Powers, Richard)
- [Unbroken: A World War II Story of Survival, Resilience, and Redemption](#) (Hillenbrand, Laura)

Local Library Views for Books

The image displays three overlapping browser windows from a local library application running on localhost:3000.

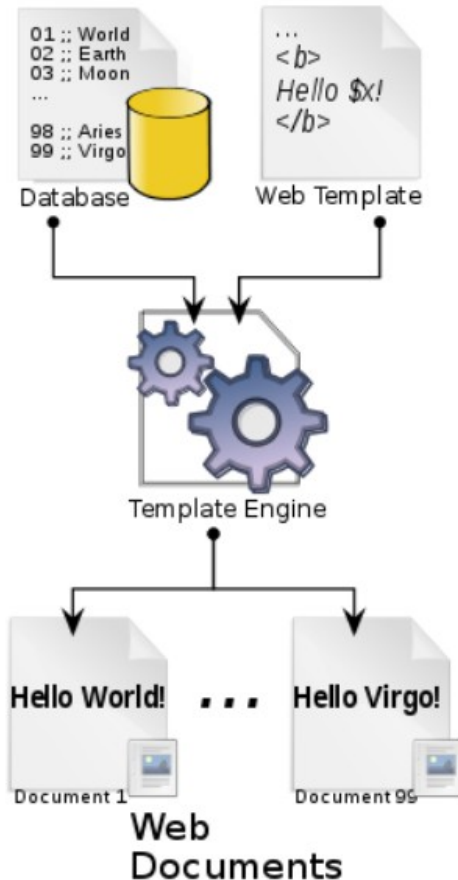
- Top-Left Window (Title: The Overstory):** Shows the book details page for 'The Overstory' by Richard Powers. It includes a sidebar with navigation links (Home, All books, All authors, All genres, All book-instances), a main content area with fields for Title, Author, Summary, ISBN, and Genre, and a 'Copies' section with a 'Loaned' status. A 'Delete Book' button is visible at the bottom.
- Top-Right Window (Update Book):** Shows the 'Update Book' form. The fields are pre-filled with the book's current data: Title: 'The Overstory', Author: 'Powers, Richard', and Summary: 'A monumental novel about re-imagining our place in the living world.' Navigation links and 'Create new' options are also present.
- Bottom Window (Create Book):** Shows the 'Create Book' form. Fields include Title (placeholder: 'Name of book'), Author (dropdown: 'Asimov, Isaac'), Summary (placeholder: 'Summary'), and ISBN (placeholder: 'ISBN13'). A 'Genre' section has checkboxes for Science Fiction, Poetry, Fiction, Historical Fiction, Children, Non-fiction, Romance, and Mystery. A blue 'Submit' button is at the bottom.
- Bottom-Left Window (Delete Book: The Prince and the Pauper):** Shows a confirmation dialog for deleting the book 'The Prince and the Pauper' by Mark Twain. It includes the book's summary, ISBN, and genre, followed by a question 'Do you really want to delete this Book?' and a blue 'Delete' button.

Local Library Views



Local Library Views use Pug

- Pug is a template engine for Node that compiles Pug code into HTML
- Templates are text files that define the structure or layout of an HTML document, and include placeholders where data will be inserted when the page is rendered
- It uses a simplified syntax and allows you to reuse static web page elements (e.g., header, footer, navigation)



```
extends layout book_detail.pug

block content
  h1 Title: #{book.title}

  p #[strong Author:]
    a(href=book.author.url) #{book.author.name}
  p #[strong Summary:] #{book.summary}
  p #[strong ISBN:] #{book.isbn}
  p #[strong Genre:]
    each val, index in book.genre
      a(href=val.url) #{val.name}
  ...
```

References:

MDN Web Docs: [Template Primer](#)

SitePoint: [A Beginner's Guide to Pug](#)

Local Library Views for Book Detail

The screenshot shows a web browser window with the URL `localhost:3000/catalog/book/5cfd35e0296b4f3fd439b53d`. The page content is as follows:

Title: The Overstory
Author: Powers, Richard
Summary: A monumental novel about re-imagining our place in the living world.
ISBN: 9780393635522
Genre: Fiction

Copies

Loaned

Imprint: Norton
Due back: Thu Apr 29 2021 20:00:00 GMT-0400 (East...)
Id: 5cfeec538e359926cce4a242

Delete Book
Update Book

`layout.pug`

```
extends layout book_detail.pug

block content
  h1 Title: #{book.title}

  p #[strong Author:]
    a(href=book.author.url) #{book.author.name}
  p #[strong Summary:] #{book.summary}
  p #[strong ISBN:] #{book.isbn}
  p #[strong Genre:]
    each val, index in book.genre
      a(href=val.url) #{val.name}
  ...
```


Part 1: Overview, Installation

Part 2: Views

Part 3: Models

Part 4: Routes and Controllers

Local Library Model uses MongoDB

- **MongoDB** is an open source, NoSQL database that uses a document-oriented model.
- Data is stored as a collection of documents in a JSON-like format that can represent dictionaries, lists, and nested structures.

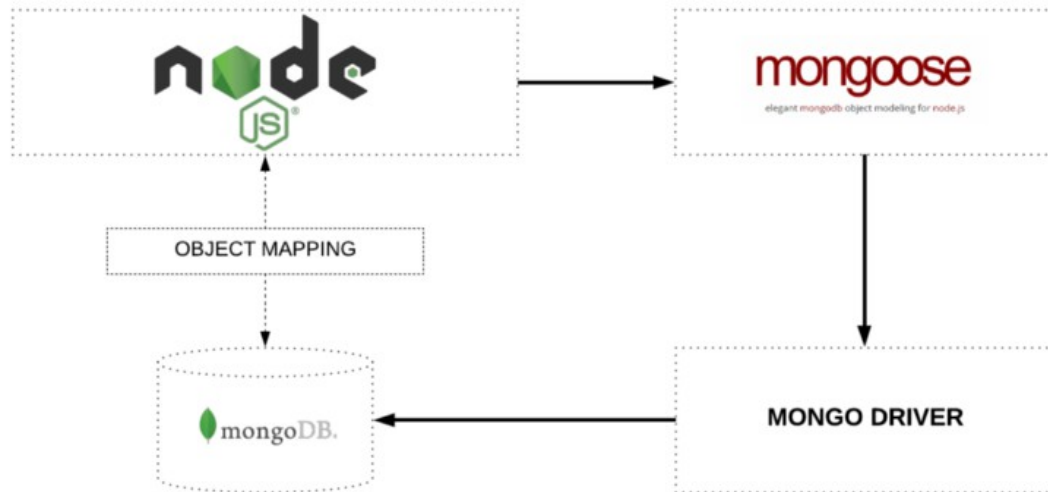
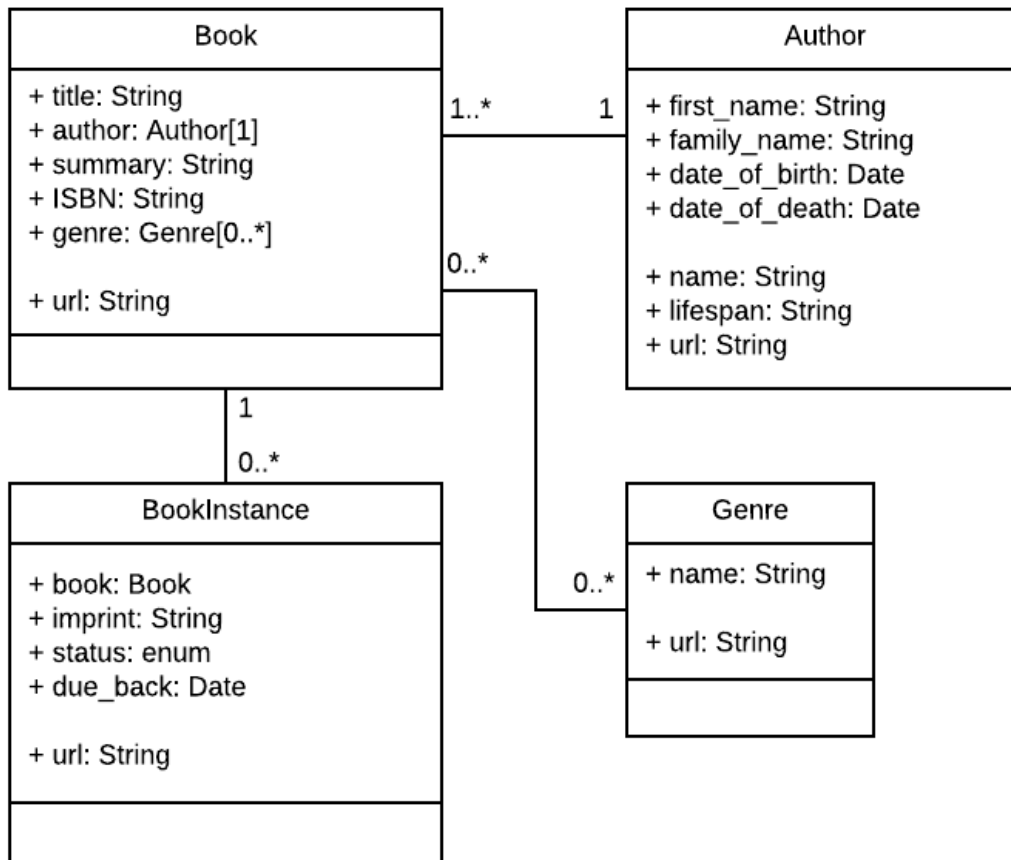


Image source: [freecodecamp.org](https://www.freecodecamp.org)

Mongoose is a data abstraction layer that

- supports portability across different database models
- provides an interface that simplifies usage by not requiring an in-depth understanding of the implementation details

Local Library Model uses MongoDB



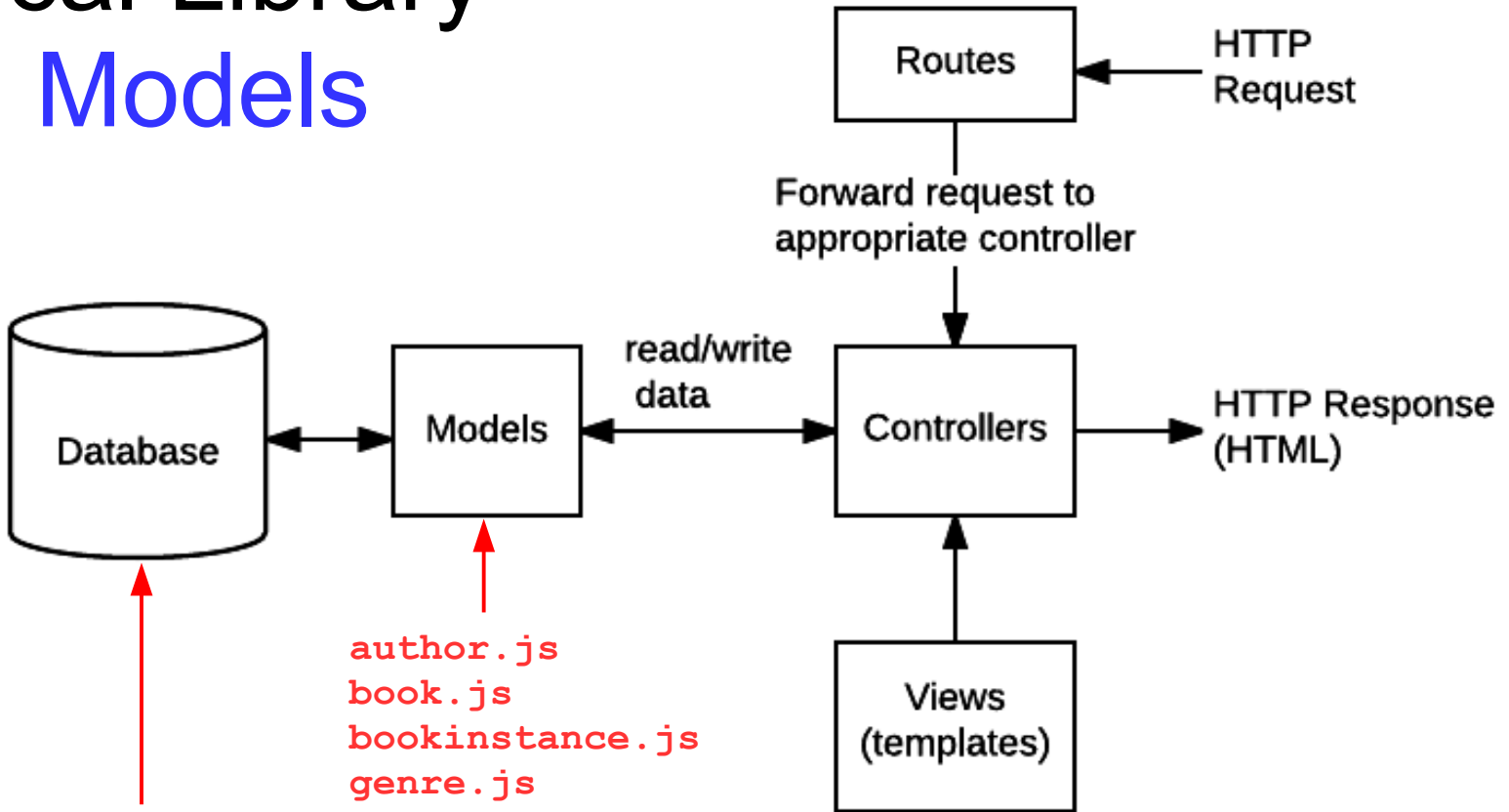
- UML diagram illustrates the models for Book, Author, BookInstance, and Genre.
- Each box defines the model name, the field names and types, and the relationships with other models.

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var BookSchema = new Schema({
  title: {type: String, required: true},
  author: { type: Schema.ObjectId, ref: 'Author', required: true },
  summary: {type: String, required: true},
  isbn: {type: String, required: true},
  genre: [{ type: Schema.ObjectId, ref: 'Genre' }]
});
```

book.js

Local Library Models



Database URL defined in `app.js`

```
'mongodb+srv://dbuser:dbpassword@cluster0-a2eax.mongodb.net/local_library?retryWrites=true&w=majority';
```

Note: this MongoDB database is hosted on the [Glitch](#) platform, so any updates you make will be available to everyone in this class.

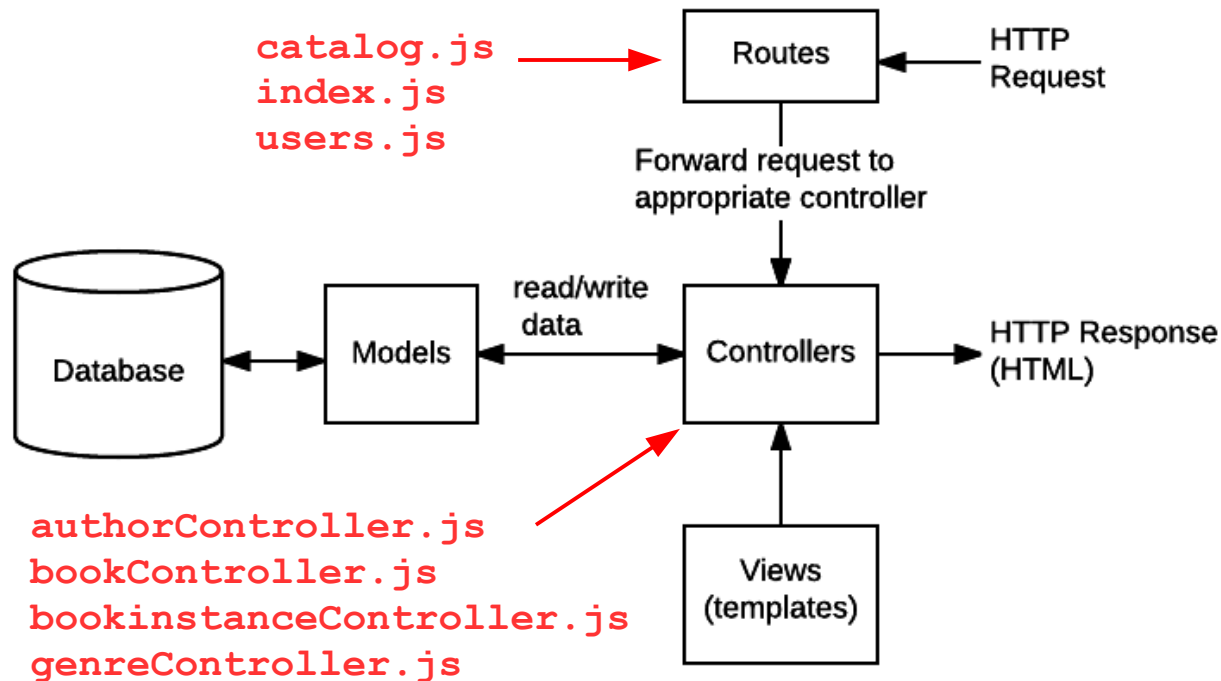
Part 1: Overview, Installation

Part 2: Views

Part 3: Models

Part 4: Routes and Controllers

Local Library Routes and Controllers



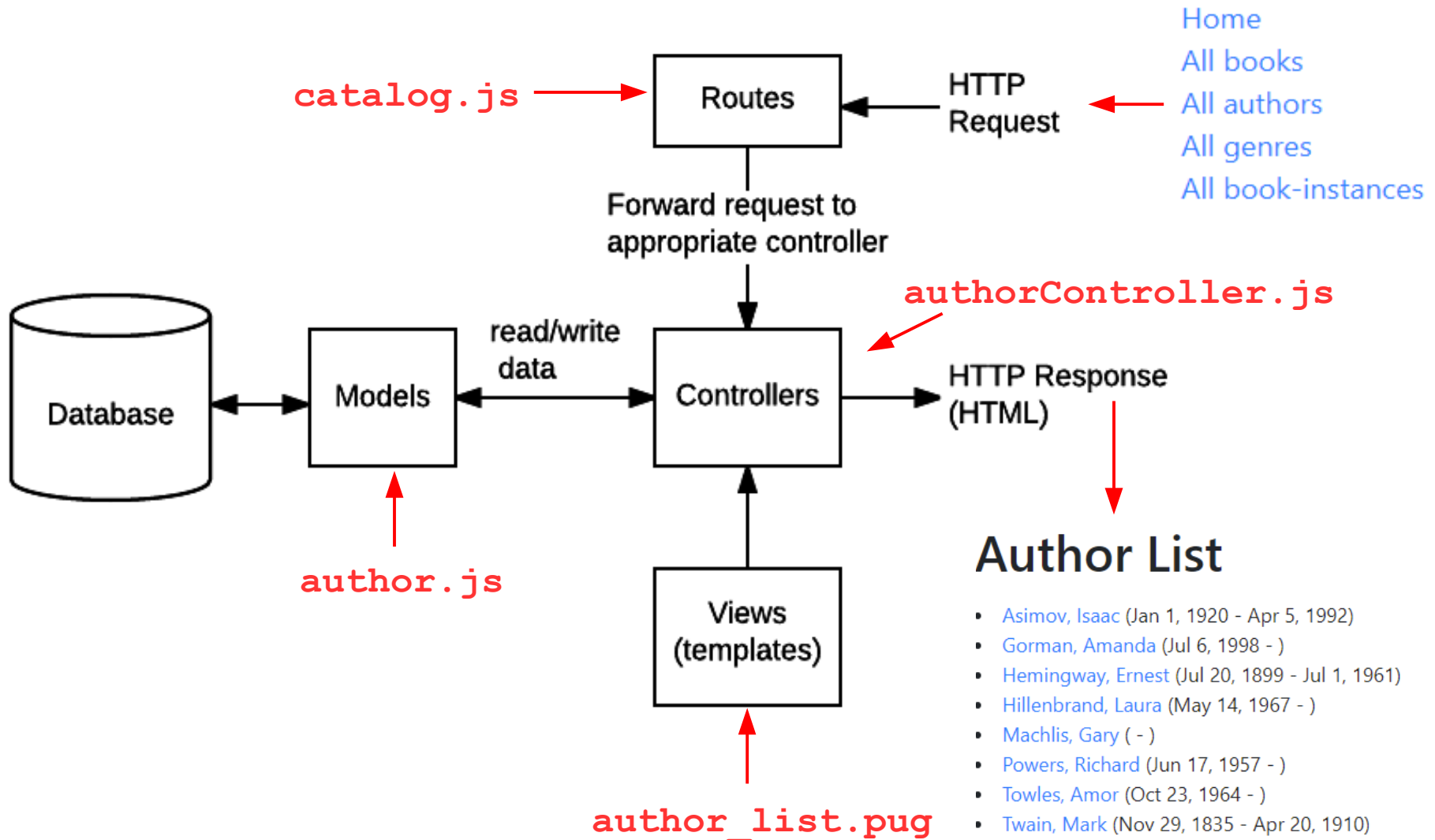
- **Routes** handle HTTP requests and determine which controller to forward the request to. There are separate controllers to handle book, author, genre, and instance-related requests.
- **Controllers** receive the requests and access the models to get or post data. Then, they obtain the appropriate View template to create an HTML page with the results, and return it to the user via an HTTP response.

Route files are located in: [express-locallibrary-tutorial-master/routes](#)

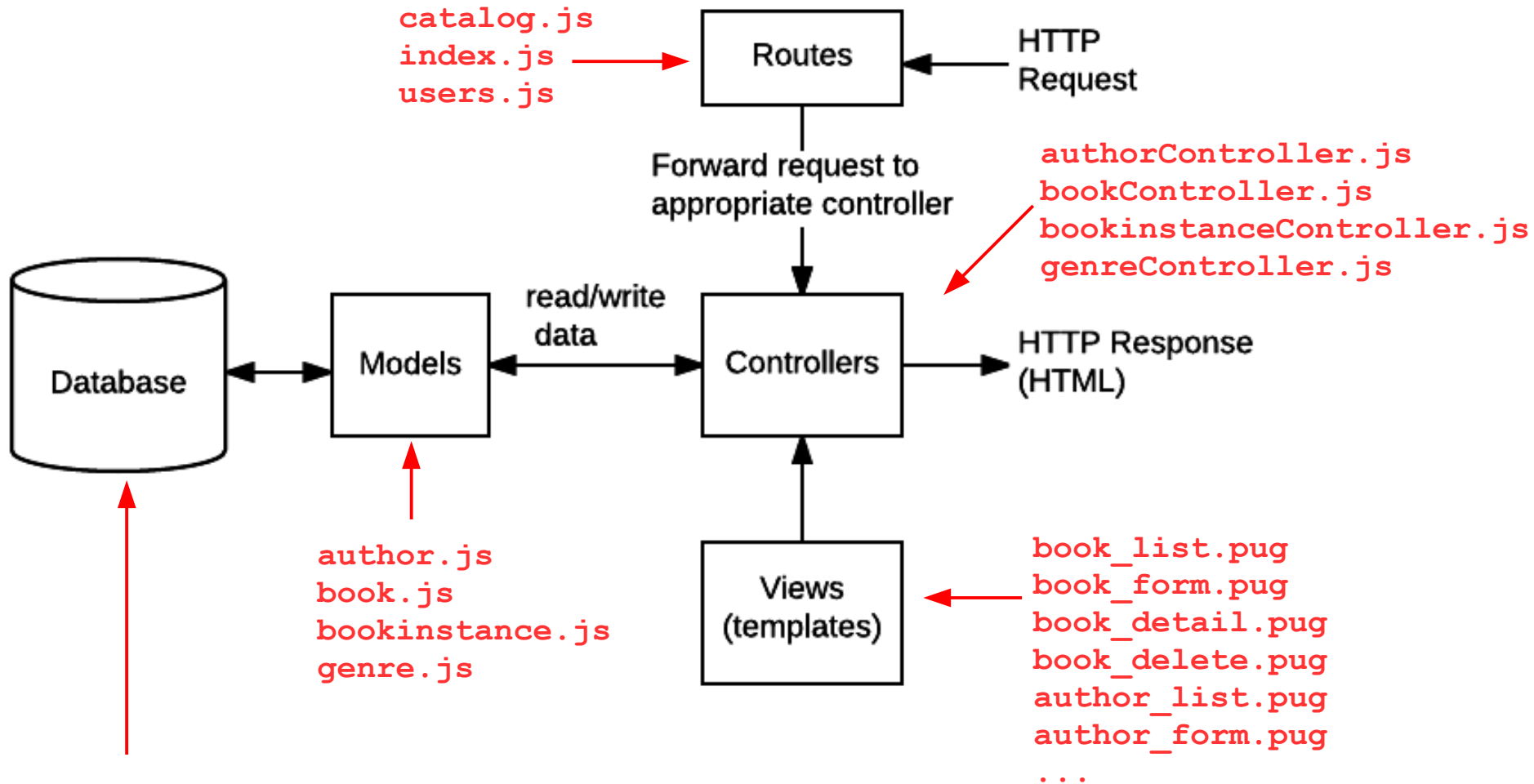
Controller files are located in: [express-locallibrary-tutorial-master/controllers](#)

MDN Web Docs: [Routes and Controllers](#)

Example: 'All authors' Request



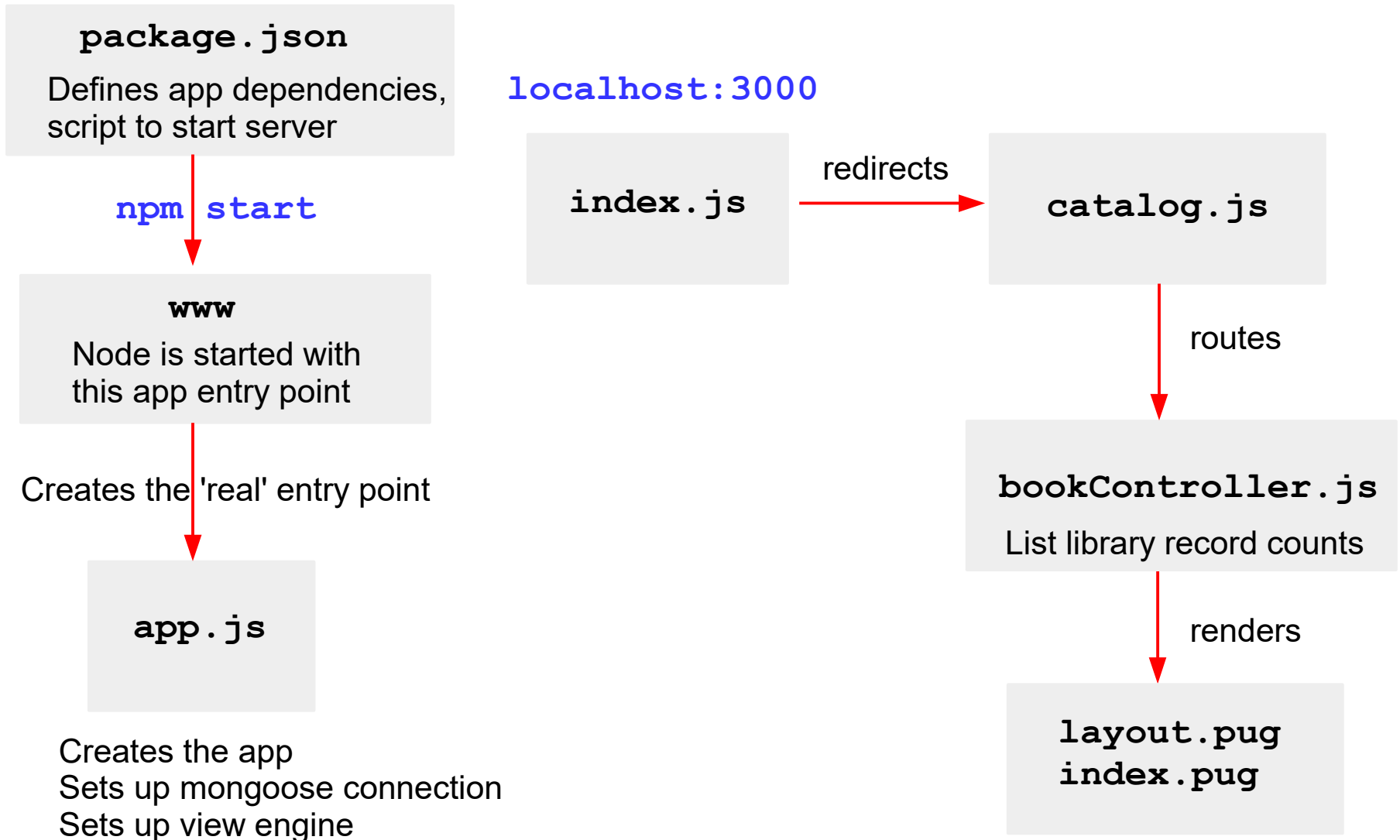
Summary: JavaScript Code for Local Library Application



Database URL defined in app.js

```
'mongodb+srv://dbuser:dbpassword@cluster0-a2eax.mongodb.net/local_library?retryWrites=true&w=majority';
```


Roadmap for Node.js App Startup



Local Library Home page at Startup

The screenshot shows a web browser window with the title "Local Library Home" and the address bar containing "localhost:3000/catalog". A red arrow points to the address bar with the text "Note URL change". The page content is divided into two main sections:

- layout.pug**: A sidebar containing navigation links: Home, All books, All authors, All genres, All book-instances, Create new author, Create new genre, Create new book, and Create new book instance (copy).
- index.pug**: The main content area with the following structure:
 - ## Local Library Home
 - Welcome to *LocalLibrary*, a very basic Express website developed as a tutorial example on the Mozilla Developer Network.
 - ## Dynamic content
 - The library has the following record counts:

 - **Books:** 10
 - **Copies:** 19
 - **Copies available:** 3
 - **Authors:** 8
 - **Genres:** 8