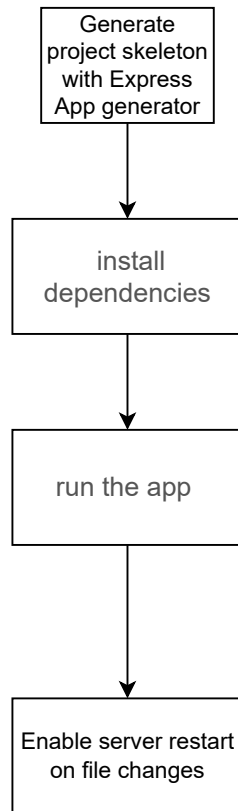


## 1. Setup app



```
npx express-generator
```

```
npm install
```

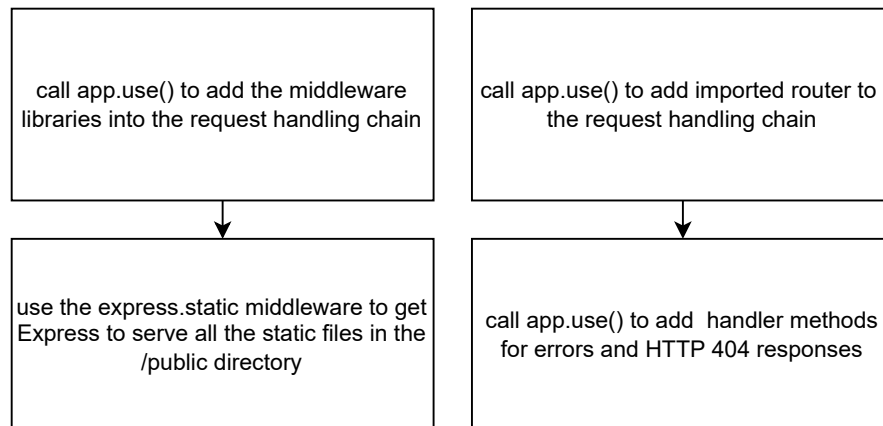
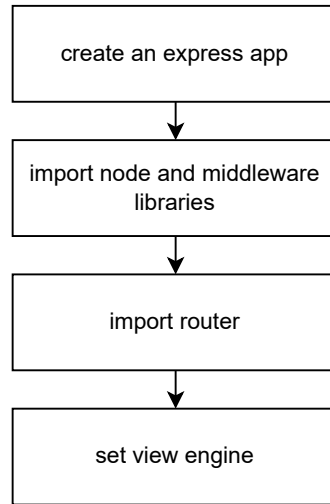
```
Linux: DEBUG=myapp:* npm start
```

```
npm install --save-dev nodemon
```

```
"devstart": "nodemon ./bin/www",
```

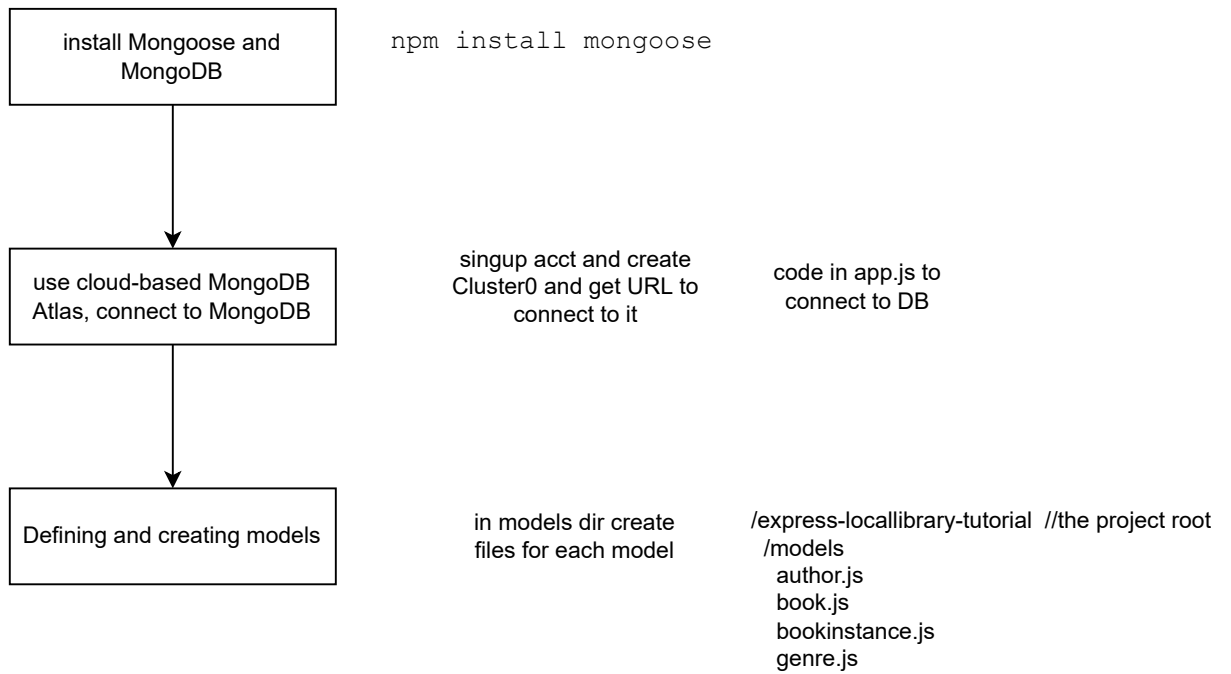
```
"serverstart": "DEBUG=express-locallibrary-tutorial:* npm run devstart"
```

## 2. app.js file

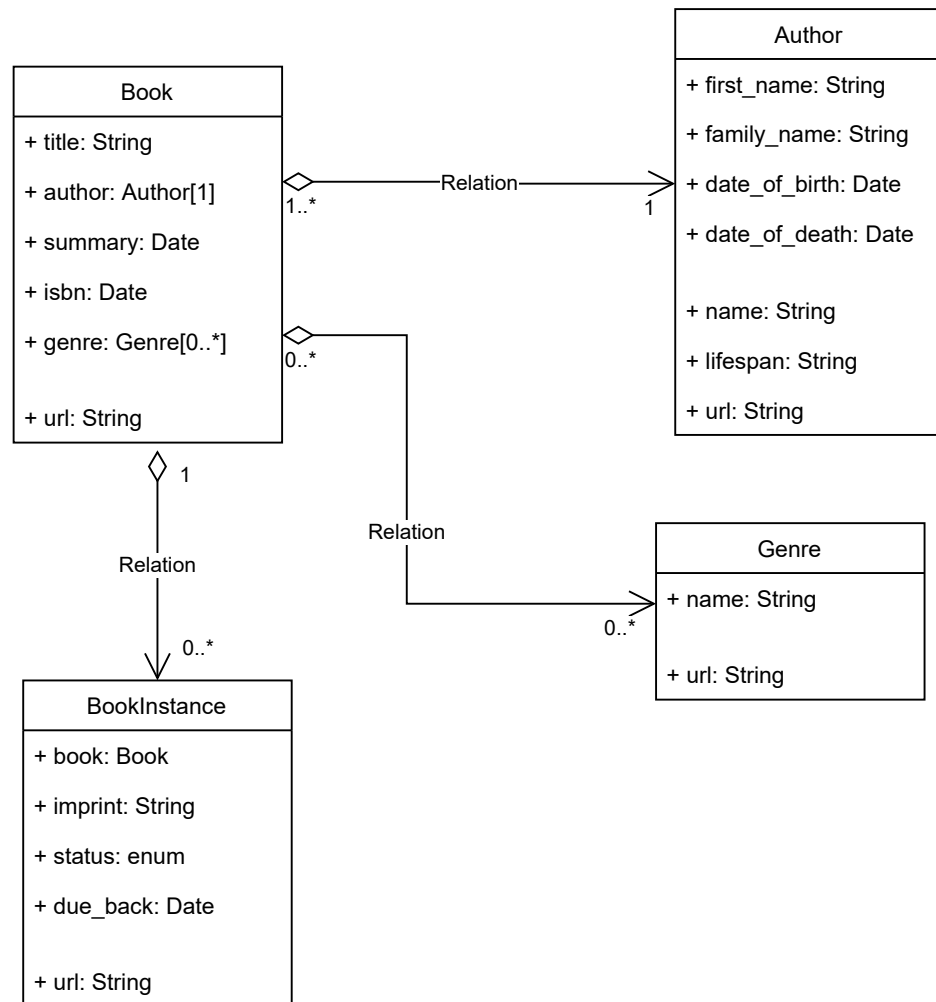


### 3. Setup DB

#### Mongoose

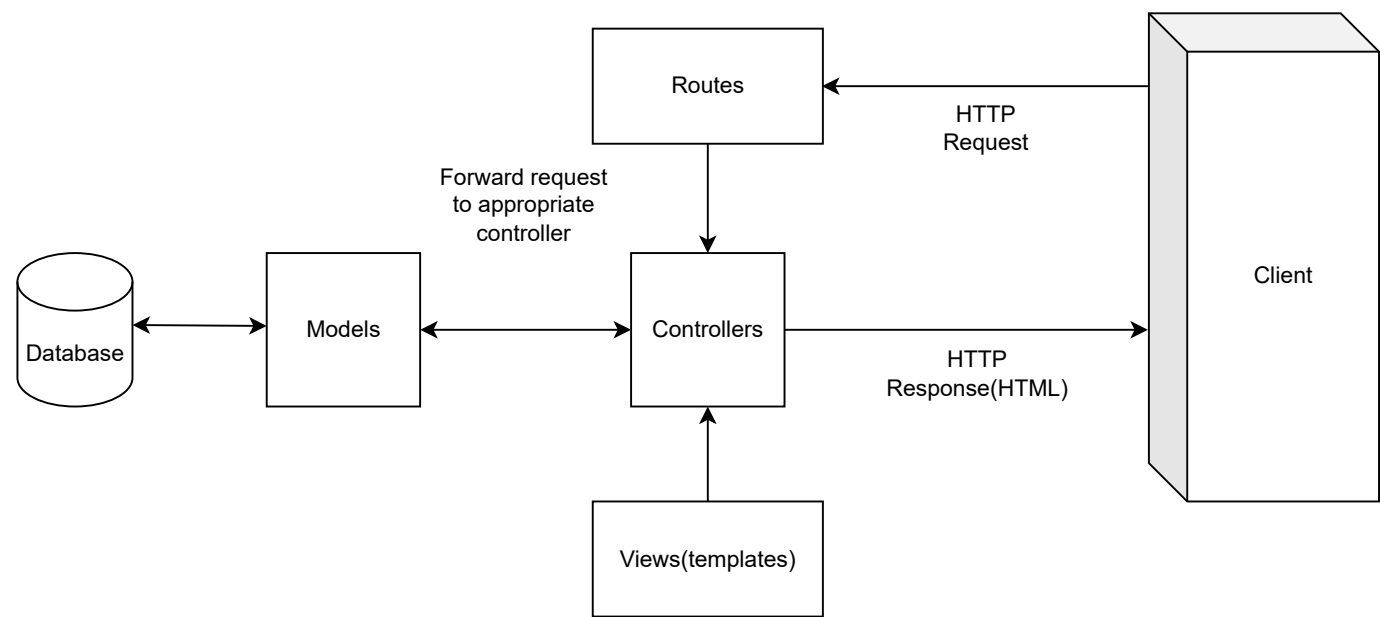


#### 4. Create model



Testing: using a script to create instances data

**Main flow of data and HTTP req/res handlers**



## RESTful API

catalog.js
catalog/book
catalog/author
catalog/genre

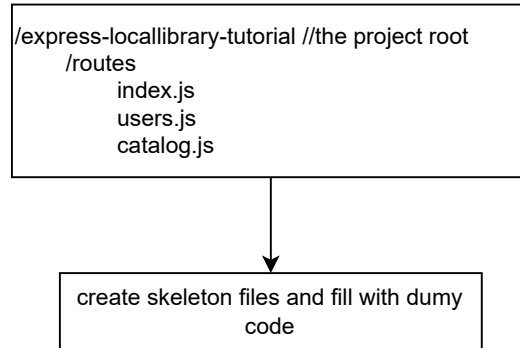
HTTP verb
GET
POST

book_controller
+ book_create_get()
+ book_create_post()
+ book_delete_get()
+ book_delete_post()
+ book_update_get()
+ book_update_post()
+ book_detail()
+ book_list()
+ book_index()

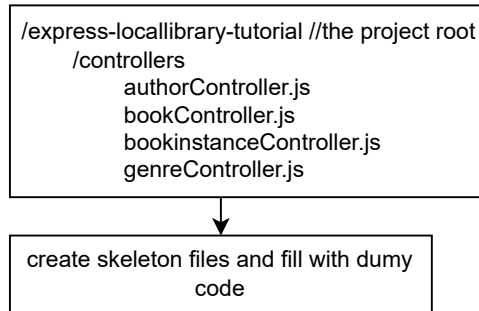
author_controller
+ author_create_get()
+ author_create_post()
+ author_delete_get()
+ author_delete_post()
+ author_update_get()
+ author_update_post()
+ author_detail()
+ author_list()

genre_controller
+ genre_create_get()
+ genre_create_post()
+ genre_delete_get()
+ genre_delete_post()
+ genre_update_get()
+ genre_update_post()
+ genre_detail()
+ genre_list()

## 5. Create routes

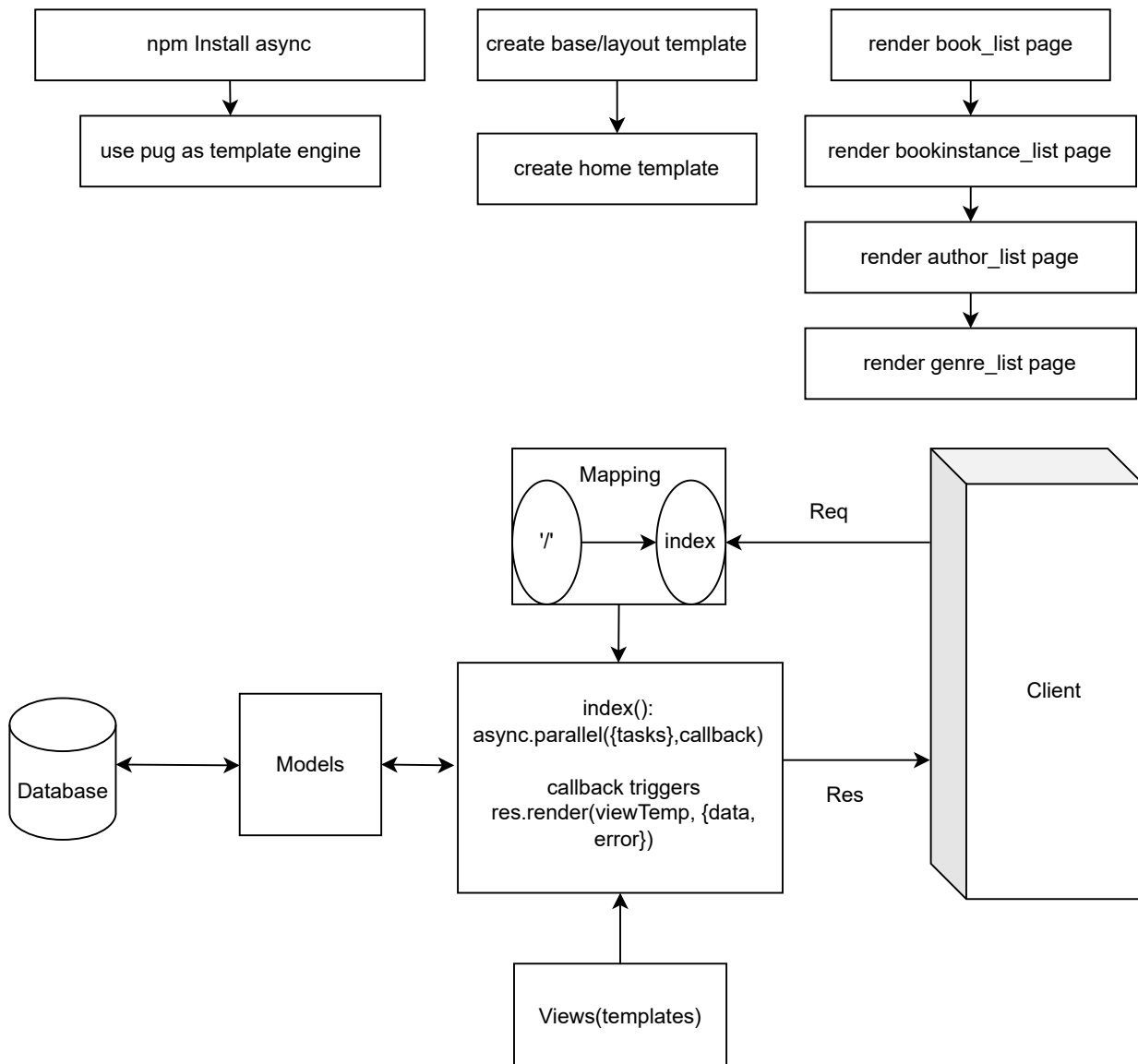


## 6. Create controllers skeleton





## 7. Config async and create templates



data obj is passed through res.render() method to create html and send back to the client

data can be from the DB(i.e models) or the req.body.params

date formatting using 3rd party library

```
npm install luxon
```

```
const { DateTime } = require("luxon");
```

```
use:DateTime.fromJSDate(this.due_back).toLocaleString(DateTime.DATE_MED);
```

## 7. Work with forms

create/update/delete document in DB

install middlewares needed to  
perform form data validation and  
sanitization

specify the function  
we want to import in  
each each controller  
file

npm install express-validator

```
const { body, validationResult } = require('express-validator');
```

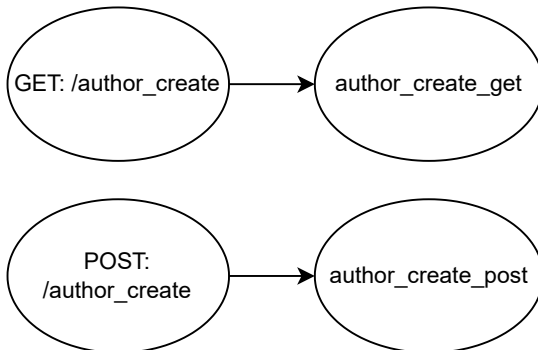
Create form

Router

Controller

HTTP  
Method

controller



author\_create\_get

res.render("author\_form", {data})

author\_create\_post

validate and sanitize data  
with middlewares and passing  
them as an array to the controller



## 8. Deploy