**MVC结构**

**1.基础设置:**

* 设置好view engine, views和layout路径(作为前端), 以及static文件夹
* 监听3000端口
* 设置路由
* 连接mongoose，有错报错，没错就console.log一次连接成功
* 配置env, 并且设置当目前模式不为production的时候，才会require dotenv并把它config()
* 建立author路由，包含全部author, 新的author，以及创建author(post)三个路由
* 前端views中建立partials文件夹(局部渲染)， author文件夹(author, new author)

**2.Author搜索页面:**

* 用mongoose建立author的模型, 在路由中拿到，传入建立新author的路由中
* 在前端构建表单，并把输入框绑定后端路由从模型中拿到的author的名字
* 安装并使用body-parser来接收请求体
* 在创建新author的路由中，设个对象赋值为请求体。 如果路由进入成功，在前端显示新加的名字。不成功就显示errorMessage
* **显示全部author/显示搜索的author：**
  + 用author模型的find({})获取并传到前端显示全部authors
  + 设个searchAuthor变量，并在req.query.name不为null和不为空的情况下把searchAuthor赋值为req.query， 让author模型find这个searchAuthor(如果搜索栏为空就还是显示全部)，并把searchAuthor传到前端并显示

**3.书本信息上传页面**

* 建立book模型，里面包含(title, description, publishDate, pageCount, createdAt, coverImageName和author)， 这个author实际上是author模型的id
* 建立book路由，和author构造差不多
* 去前端建立books文件夹，建立和author构造一样的new.ejs，同时在这文件夹下建立相应的\_form\_fields.ejs

\_form\_fields.ejs

* + Title 输入框
  + Author下拉选项：遍历authors, 从中拿到的author名字就是它的label，拿到的id就是它的value。 默认选中为当前author的id

Book路由：

* + 添加新书的路由: 分别拿到Author.find({})的所有对象和Book模型，并把它们以一个对象形式render到前端
* 到前端把其它输入框渲染好
* 完成表单整个添加:
  + New book的路由封装到另一个renderNewPage函数中，传入res, book, hasError(默认为false)， 如果hasError是true, 那就显示错误信息
  + 在post book的路由中，把全部需要在前端显示的传到前端，并把它们save到一个新对象中
  + Book model中配置存放图片的路径并export
  + 回到book路由中，安装并使用multer, 使用path, 拿到uploadpath，定义image type， 并用multer把uploadpath和一个filefilter函数
* 如果表单有地方没填，还是会把图片保存，防止这种情况:
  + Require fs, 在catch中如果存在新加入的coverImage, 就建立并调用removeBookCover函数
  + removeBookCover: unlink这个file

**4.书本搜索页面**

* 前端渲染搜索页面(index.ejs), 表单下面渲染书的coverImage( src拿到book模型的coverImagePath);
* 在book模型中用bookSchema建立一个virtual方法， 用get()来返回一个函数，这个函数返回path.join(‘/’, 图片路径，图片名)
* 到book路由的获取所有书的路由里，把所有书和搜索结果传到前端，用query获取搜索的值
* 最后把目前所有的书显示到前端的index.ejs
  + 在index.ejs的后端路由中把book传到前端，排序为最新的在前限制10本

**5. 图片上传优化**

* 去到file pond上cdn导入主css和script文件，再倒入image preview的js和css, file encode和image resize的js（这些都要放在主js前面）,再建个名字为fileUpload的js文件，用来引入这些功能
* 在fileUpload中，FilePond.registerPlugin方法分别传入preview,resize和encode方法。去到上传书图片的前端页面中，加入class=’filepond’
* 再回到fileUpload中，.setOptions（）里面定义图片比例，和长宽尺寸