С req.body взимаш данните от **form**

Когато правиш заявка с find() добавяй .lean() накрая, за да вземеш данните като обикновен обект.

Params е частта от URL, от която взимаш данни

Ако подаваш повече опции, напр. title, при render на view трябва да се подават в обект, а във view-то темплейтът да извиква пропъртита на обекта.

При catch на асинхронна функция, error се подава в обект, защото така се подава при грешка от Node

ВНИМАВАЙ КАК РЕФЕРИРАШ STYLE.CSS в main.hbs(html)– ВИЖ В CUBICLE КАК Е РЕФЕРИРАН!! ТОВА, КОЕТО Е ЗАДАДЕНО В EXPRESS.STATIC Е НАЧАЛОТО НА PATH И СЛЕД ТОВА ПРОДЪЛЖАВА С /CSS/STYLES.CSS И ТРЯБВА ДА Е С href!!!Сложи / в началото, т.е. след static(или public) – името на папката, в която са файловете.

1. ~~Задаваш данните в config.js~~
2. ~~Инсталирай нужните библиотеки –~~ **~~предварително~~**
3. ~~Създай database –~~ **~~предварително~~**
4. ~~Създай папка layouts & partials – може и~~ **~~предварително~~** ~~да създадеш файл main, в който ще слагаш всичко, което е в main – ВНИМАВАЙ С ТРИТЕ {{{body}}}~~
5. Преименувай всички html файлове на ‘hbs’, внимавай с имената при routing
6. Раздели темплейтите на main и останалите в отделни темплейти
7. Сложи {{title}} в title tag-a в main
8. Сложи paths където е необходимо в hbs темплейтите
9. Внимавай дали в темплейтите inputs имат name - защото така взимаш данните с req.body!!
10. Не прави error handling & authentication в началото, **а накрая – в зависимост от точките, които дават**
11. Задай динамичните части в handlebars темплейтите

Виж за по-свестна валидация в Theaters project – user & product models

В package.json – dependencies е махната чавката пред версията на всички npm пакети, ако има проблем я сложи – в exam prep 16.02 на 1.47.00 има инструкции!!!