

Little about requirements and why is something done like it is and what could be improved :)

Creating table and storing records - I saw that **dropwizard** have this feature so if I am using dropwizard I wanted to try to use most of his features.

Validations - I didn't want to use annotations because in this way I can customize response and parse it on front(probably it can be done by annotations but this is familiar way for me, in company we validate parameters like this). This should be done much better, we can send just ERROR ID and then on front parse text for this ERROR ID.

For **Home page** I created GET method to return last five inserted records, I saw in requirements that client should read a few of them so for any case I did in this way.

Get all books API wasn't part of requirements but I did it for testing and then leave it, this can be easily removed.

Get book – I did this as part of requirement “read information about specific book”, it just check for title with LIKE clause, this could be done with QueryParams too, to search by authors, genre but wasn't specified

For **database** I wanted to create separate table for authors but this would take additional time:)

About **frontend** part, I used **jquery, html, css** (I needed to learn this again:)) I didn't know can I use **npm, bower, knockout** ... so I did it in simple way. This can be improve a lot but for now just showing that APIs are working and displaying some errors. There are some **validations** on **frontend** part and most of them on the **backend** (motto is “never trust the client”) so for just testing **APIs** please use **postman** collection from postman folder in root of project.