Express Assignment

Write a REST API servicer in Express.js to track library books with the following properties:

- id (string)
- title
- author
- publisher
- isbn
- avail (true/false, false means it's checked out)
- who (has it checked out)
- due

GET requests should return JSON. PUT/POST requests should include JSON. Use these exact property names in all JSON No database is required -- your service handle data via standard JavaScript objects & arrays, similar to my pet server example.

Initialize your inventory of books with the following data:

id	title	author	publisher	isbn	avail	who	due
1	Reactions in REACT	Ben Dover	Random House	978-3-16-148410-0	true		
2	Express-sions	Frieda Livery	Chaotic House	978-3-16-148410-2	true		
3	Restful REST	Al Gorithm	ACM	978-3-16-143310-1	true		
4	See Essess	Anna Log	O'Reilly	987-6-54-148220-1	false	Homer	1/1/23
5	Scripting in JS	Dee Gital	IEEE	987-6-54-321123-1	false	Marge	1/2/23
6	Be An HTML Hero	Jen Neric	Coders-R-Us	987-6-54-321123-2	false	Lisa	1/3/23

Your server should support the following operations:

Method	URL	Query	Function
GET	/books		returns a list of all books (title & id) regardless of checkout status (i.e. both available and checked-out books should be included)
GET	/books/id		returns all details for the book matching id, 404 if not found
GET	/books	avail=true	returns list of all books (title & id) which are available to be checked out
GET	/books	avail=false	returns list of all books (title & id) which are NOT available to be checked out
POST	/books		Add a new book as described in request body (JSON), which includes id & status
PUT	/books/id		For book matching id, update any properties present in request body (JSON). Can be used to check in or out.
DELETE	/books/id		Delete the book matching id (if it exists) regardless of checkout status

These are the http response codes I expect:

CRUD Action	HTTP Method	ID Exists?	http status
Create	POST	Yes	403 already exists
Create	POST	No	201 created
Read	GET	Yes	200 ok
Read	GET	No	404 not found
Update	PUT	Yes	200 ok
Update	PUT	No	404 not found
Delete	DELETE	Yes	200 ok
Delete	DELETE	No	204 no content

Request & response bodies should use JSON where appropriate.

Your server should handle CORS via Rickster's minimal CORS handler below (or something similar):

```
app.use(function(req, res, next){
  res.header('Access-Control-Allow-Origin', '*');
  res.header('Access-Control-Allow-Methods',
'GET,PUT,POST,PATCH,DELETE,OPTIONS');
  res.header('Access-Control-Allow-Headers',
       'Content-Type, Authorization, Content-Length, X-Requested-With');
  if (req.method === "OPTIONS") res.sendStatus(200);
  else next();
});
```

Don't use any middleware or other dependencies. This requirement is just to simplify the effort involved in running your project on another computer.

Your server should run on http://localhost:3000

Submission options:

- If your entire server is contained within one .js file, you may submit just that file.
- Alternatively, identify which file should be started and submit a .zip of the folder containing that file, ensuring that all other files are within that tree.