

For the following tasks please use the attached JSON file as data source.
You may use any package available on npm.
You can name the endpoints and other parts of the tasks to your liking unless stated.
Make your code available in a private repository either on github or bitbucket.

Task 1

Implement the common CRUD endpoints for the resource Rental. You do not have to update the source file, but respond with data that makes sense (e.g. sending back any of the existing Rentals when trying to create a new one). Treat every field except *views* as required when creating a new or updating a resource.

Task 2

Implement JWT authentication for all endpoints. You can find a list of dummy users in the users.json file. Authorization for specific rentals is not required. For the sake of simplicity the passwords in the dummy file are neither hashed nor salted. Also provide an authentication endpoint. Respond properly to invalid authentication credentials in the authentication endpoint as well as in all other endpoints.

Task 3

Log any HTTP error response in a log file. The file should be in CSV format including the date, the response status code and an error message.

Task 4

Serve all files inside the xy directory under the path prefix static (e.g. file example.pdf should be available as /static/example.pdf). The user does not have to be authenticated to get these static files.

Task 5

Wrap every response in an object of the following format:

```
{
  "status": 200, // HTTP status code
  "data": {} // The actual payload
  "message": "Success" // Or error message
}
```

Task 6

Implement an endpoint to upload files to a specific directory on the server. The target directory should be configured via environment variables. Only allow PNG graphics to be uploaded to that directory.