

Backend Task

We'd like you to build a URL shortening API (similar to <https://tinyurl.com/app>, don't worry about building the UI, we are focused on the API). This service should be able to take a long URL e.g. "https://foobar.com/my_super_long_url_that_no_one_remembers" and return a unique shortened URL that when accessed will redirect the requester to the original long URL.

When a user accesses the short URL you should record a set of statistics for its use against the original URL. These statistics will later be used to "score" the stored short URLs based on how many times they are used. The API should be built using the Nest framework (you do not need to host the API) in NodeJS. Data should be stored in some form of data store. This can be a hosted/local database, an in memory data store or just a local JSON file.

Please ensure you enforce a consistent coding style. You should include tests to ensure the functionality and consistency of the API you build.

Task

1. Build an API endpoint that accepts a long URL as its input and ultimately returns a short URL that when accessed will redirect the user to the original long URL
2. When the short URL is accessed you should store statistics about its use against the original URL. Think of this as a scoreboard for all of the short URLs, how many times are they accessed, where are they accessed from, by what unique users
3. Build an API endpoint that will return the URLs and their stored statics. Just imagine we have a UI that will present a list of URLs alongside its statistics like a scoreboard
4. Provide API documentation that conforms to the OpenAPI standard
5. Dockerize the project

Bonus

If you have time, implement an API endpoint that allows a user to modify their short urls.

1. The ability to set an alias for the auto generated short url e.g. `shorturl.com/123456789 > shorturl.com/myshorturl`
2. The ability to add a request limit to the URL, perhaps you only want the first 10 people to who request the short URL to be redirected to the original url

3. The ability to delete a short url. Think about the best way to let users know that a shortURL they are accessing has actually been deleted

Extra bonus

Think about concurrency and API performance. What happens if the API is receiving thousands of requests per second? How can you enforce data integrity and ensure that the API is responding promptly to its users?

Submission

Please push your code to a public github repository and share the link with us. Make sure to include the necessary instructions for us to run the nest project as well as anything else you think we should know. Code comments on areas of interest are encouraged, help us to understand why you have developed something in a particular way.