A Simple URL shortening API

Functional Requirements

Design and implement a simple service to shorten urls. At a high level the service should

1. Take a url, validate the url and return a `json` object containing the id of the shortened url as well as the shortened url
2. Expose an endpoint that will redirect a user from a shortened url to the real url
3. Expose an endpoint that can use the hash of the url to look up and return information about the url including its id, the shortened url and the original url

- Shorten URL Endpoint
    - Endpoint takes a url as input, stores the full and shortened url in the db and returns a JSON payload

```
{
      "url_to_shorten" : "http://somewebsite.com"
}
```

returns

```
{
      "id" : 1234,
```

```
    "shortened_url" : "http://shorten.io/sY4"
}
```

- Redirect shortened to full url :

  `http://shorten.io/sY4` redirects to `http://somewebsite.com`

- Retrieve Shortened URL info by ID Endpoint
  - `http://shorten.io/urls/sY4` returns

```
{
    "id" : 1234,
    "shortened_url" : "http://shorten.io/sY4",
    "original_url" : "http://somewebsite.com"
}
```

Stretch Goal

- Using your endpoints, implement a UI where the user can submit a url and that presents the user with a shortened link to that url.
- Provide a WebSocket interface

Non-Functional Requirements

- The service must be implemented in `node.js`
- The service must be backed by a `Postgres` database

- The service must be hosted in AWS and accessible online

Things to keep in mind

- As always design and write code that is testable, extendable and easily understood by your teammates
- As always when designing a database schema keep data-integrity and scalability in mind
- Feel free to use an api framework and a Postgres client of your choosing and be prepared to discuss why you choose to use the tools you did
- Don’t worry about implementing production level features like authentication, metrics, logging and other cross-cutting concerns. Do worry about creating a system (or choosing a framework) that allows for those cross-cuts to be easily plugged in at a future iteration
- Have fun! Take this as an opportunity to make something beautiful, the task is a simple one - we trust that most candidates will be able to produce a working solution to this problem. Thus, we will be evaluating your aesthetics, judgement and the elegance of your solution.
- Feel free to ask us questions!