Project - Bookmark & URL Shortener APP V2

(To better understand the features to be created in this version, setup an account on bit.ly and analyse the working of it)

In this version, we will add on top of our existing Application.

New Features

Add clicks as a field to the existing Bookmark Schema. This field is used to store the collection of click information that we require for future statistics and data mining.

Note - A document field can hold very complex set of data within itself, here the click field will hold an array of objects. To learn how to define it in the schema check out this link and learn and understand how the blog schema has been created

Each click object will store the following information

- Clicked date and time - should be set to default datetime
- Ip address of the user - must be stored as a string
- Browser Name - must be stored as a string
- Os type - Machintosh / Windows / Linux - must be stored as a string
- Devise Type - mobile / desktop - must be stored as a string

Tip - Each request object will have the above mentioned details, read more here.

In this project implement the express-useragent npm package for the getting the details of the request.

API endpoint for SHORT URL

HTTP Method	URI	Actions
GET	/:hash	Finds the url with the hash value and redirects the user to the respective page

When a specific url is accessed, push the object containing above mentioned information into the clicks array by using the $push method

Setup an 404 handler

What if the user tries to access a route that doesn't exist within your application.
Express defaults to an error which says "Error - route not found".

Eg - http://localhost:3000/add/url

Instead you are expected to create middleware function which will respond with a status code of 404 and send back json with notice "The resource you are looking for doesn't exist."

Know more about the error handling middleware here.

Logging the requests

1. Implement logging within the application, such that whenever the request is made to your node application, it console logs the message. Implement morgan package within your application and set it up as a custom middleware.

2. The logs should not only printed on the console, but it should also be written to a file, that will be under project_root/logs/access.log. Did you know that node provides modules for accessing the file system on you computer. Read about node fs module before proceeding.