

# Homework 5: Node.JS

In this assignment, you will develop a set of REST APIs capable of reading a JSON file deployed on a server and returning information by using a collection of simple HTTP requests. You will also build a single HTML (twitterRest.html) web page in which the requests to the APIs will be invoked and the retrieved information will be displayed.

## Overview

Companies like Twitter have strong RESTful APIs that allows users to explore the wealth of data they manage and even provide ways to download your entire archive of tweets. Many users rely on Twitter's favorites as a mechanism to bookmark tweets that link to interesting content elsewhere on the web. However, in order to avoid a potential downfall in the future, where those favorites could one day become unavailable, you've decided to retrieve a list with your favorite tweets as a JSON object. A sample JSON file is posted on Canvas.

## Specification

Your RESTful APIs should be able to respond, with the appropriate data, to the following requests:

- Get all tweets (create time, id, and tweet text) available in the archive.
- Get all known Twitter users included in the archive.
- Get a list of all external links (all links that appear in any field of a tweet. Use regular expressions) included in the tweets from the archive, the links should be grouped based on tweet ids.
- Get the details about a given tweet (given the tweet's id).
- Get detailed profile information about a given Twitter user (given the user's screen name).

## Requirements

1. Install Node.JS on your Amazon instance
2. The system will consist of two main components: the front end, which should be implemented as ONE web page called index.html and the back end, which should be implemented as a Node.js server implementing the REST APIs.
3. The web page should display any usage instructions, as well as links or buttons that trigger the proper HTTP Requests on your server and then display the retrieved data.

4. Use Ajax to only update some part of your page, as opposed to refresh the full page.
5. All the operations need to be performed by the server and can only be requested via your own REST API. This means that aside from loading the initial index.html page, there should be no other pages served by the server.
6. The Node.js server does NOT require a database. For the purpose of this assignment it is ok to lose all data when the server is stopped.
7. Your REST API will read data from the JSON file provided, which contains a number of tweets. You are not going to pull data from the actual Twitter API.
8. The frontend must be accessible using the root URL.

## **Deliverables**

Your assignment should be submitted as a zip file on Canvas:

- twitterRest.zip:
  - The html file along with any CSS files.
  - JavaScript file(s) included in the index.html page and for your Node.JS server.
  - A instructional README file with the URL of AWS instance to access your Node.JS server and a list of HTTP Requests expected to be used to test your server.
- Finally, all deliverables should be neatly formatted, readable, and be properly documented (aka. have good code comments).