



COMP5347: Web Application Development

Week 10 Tutorial: Restful Service

Learning Objectives

- Understand how to create and consume restful service in Node.js

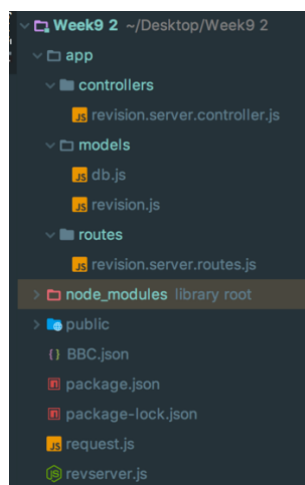
Part 1: Add request module

Task: Download the provided `week10-src.zip` from Canvas and extract the content in a directory. Start Eclipse, select “open projects from file system” under file menu, go to the directory you just extracted the content and click open. In this case, all the necessary node modules are specified in `package.json`. Right click it on the project explorer panel, select `Run as` then `npm install`.

Optional: Just keep in mind, in your own group assignment, you need to run the following command in the terminal to install “request” manually.

```
npm install request --save
```

Save the file and right click it on the project explorer panel to `Run As` then `npm update`.



Part 2: Create Restful API

Task: Start the mongodb server using the following command, replace the “path” with a path to the local mongodb working directory where you create and use in Week 7 to Week 9 (the wikipedia database with the revisions collections).

```
mongod --dbpath path
```

Right click `reverser.js` on the explorer panel and Run as Node.js application. Open the browser and go to the address below. You should be able to see something looks like the screenshot below.

<http://localhost:3000/revision/collection/BBC>

```

[{"_id":"5ca9ab333dc101ba4843122e","revid":"891210765","parentid":"891210626","minor":true,"user":"Gatmansgc","userid":"1728614","timestamp":"Sat GMT+1100 (AEDT)","size":154523,"sha1":"d0e99f7e0291c403b60742e8e95549a2f8a682d9","parsedcomment":"Reverted edits by <a href=\"/wiki/Special:Contributions/58.174.91.33\" title=\"/wiki/Special:Contributions/58.174.91.33\">58.174.91.33</a> ( <a href=\"/wiki/User_talk:58.174.91.33\" title=\"/wiki/User_talk:58.174.91.33\">talk</a>) to last version by Donner60","title":"BBC"}, {"_id":"5ca9ab333dc101ba4843122f","revid":"891210626","parentid":"891163544","minor":false,"user":"58.174.91.33","anon":"yes","userid":0,"timestamp":"23:11:54 GMT+1100 (AEDT)","size":154513,"sha1":"27d882e1d96b9e7d9f338412dc2cf05b19101ac","parsedcomment":"","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431230","revid":"891163544","parentid":"891163250","minor":true,"user":"Donner60","userid":"12744454","timestamp":"Sat GMT+1100 (AEDT)","size":154523,"sha1":"d0e99f7e0291c403b60742e8e95549a2f8a682d9","parsedcomment":"Reverted edits by <a href=\"/wiki/Special:Contributions/Trhfdthrtth\" title=\"/wiki/Special:Contributions/Trhfdthrtth\">Trhfdthrtth</a> ( <a href=\"/wiki/User_talk:Trhfdthrtth\" title=\"/wiki/User_talk:Trhfdthrtth\">talk</a>) to last version by Donner60","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431231","revid":"891163250","parentid":"890909837","minor":false,"user":"Trhfdthrtth","userid":"36265561","timestamp":"Sat GMT+1100 (AEDT)","size":154526,"sha1":"bab8ffc43ef546daeb0b967bdf3946ce710bb2b","parsedcomment":"","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431232","revid":"890909837","parentid":"890909608","minor":true,"user":"KH-1","userid":"21857263","timestamp":"Thu Apr 04 2019 21:53:11 GMT+1100 (AEDT)","size":154523,"sha1":"d0e99f7e0291c403b60742e8e95549a2f8a682d9","parsedcomment":"Reverted edits by <a href=\"/wiki/Special:Contributions/BLEH_I_DONT_WANT_VEGGIES_WAHHH\" title=\"/wiki/Special:Contributions/BLEH_I_DONT_WANT_VEGGIES_WAHHH\">BLEH_I_DONT_WANT_VEGGIES_WAHHH</a> ( <a href=\"/wiki/User_talk:BLEH_I_DONT_WANT_VEGGIES_WAHHH\" title=\"/wiki/User_talk:BLEH_I_DONT_WANT_VEGGIES_WAHHH\">talk</a>) to last version by BT","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431233","revid":"890909608","parentid":"890859946","minor":false,"user":"BLEH_I_DONT_WANT_VEGGIES_WAHHH","userid":"36253651","timestamp":"Thu Apr 04 2019 21:53:11 GMT+1100 (AEDT)","size":154510,"sha1":"1ccdb44d1b8e3105b510c2b7f02fe3069ea9","parsedcomment":"","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431234","revid":"890859946","parentid":"890858233","minor":false,"user":"Duffy BT","userid":"36210017","timestamp":"Thu GMT+1100 (AEDT)","size":154523,"sha1":"d0e99f7e0291c403b60742e8e95549a2f8a682d9","parsedcomment":"second paragraph","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431235","revid":"890858233","parentid":"890858025","minor":false,"user":"79.76.129.207","anon":"yes","userid":0,"timestamp":"11:30:34 GMT+1100 (AEDT)","size":154673,"sha1":"8bc81829bb5906f7cd632a679ee28be758701af","parsedcomment":"","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431236","revid":"890858025","parentid":"890818744","minor":false,"user":"79.76.129.207","anon":"yes","userid":0,"timestamp":"11:28:23 GMT+1100 (AEDT)","size":154578,"sha1":"8794866c03aa73a9e3049bde87594db27fc7670d","parsedcomment":"","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431237","revid":"890818744","parentid":"890818695","minor":true,"user":"Bolt Strike","userid":"35875814","timestamp":"Thu GMT+1100 (AEDT)","size":154523,"sha1":"d0e99f7e0291c403b60742e8e95549a2f8a682d9","parsedcomment":"Reverted 1 edit by <a href=\"/wiki/Special:Contributions/2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396\" title=\"/wiki/Special:Contributions/2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396\">2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396</a> ( <a href=\"/wiki/User_talk:2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396\" title=\"/wiki/User_talk:2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396\">talk</a>) to last version by BT","title":"BBC"}, {"_id":"5ca9ab333dc101ba48431238","revid":"890818695","parentid":"890497856","minor":false,"user":"2A00:23C4:3E40:9500:B0F4:4F9E:E46E:A396","anon":"yes"}]

```

Check `RevisionSchema.statics.getByTitle` in `app/models/revisions.js`. A simple callback function is defined here to query to database with given article title. In `app/controllers/revision.server.controller.js`, a controller is defined to return the query results as JSON objects. Overall, a REST API is created that accepts GET request with given article title and returns the results in JSON format.

Part 3: Consume REST Service

In this part, you are required to implement a parser which can parse the JSON objects returned from REST API and output the associated results in console. After implemented, when you run `request.js` as Node application, it should print like this in the console:

```

url: https://en.wikipedia.org/w/api.php?title=CNN&rvstart=2020-03-01T10:00:00Z&rvdir=newer&action=query&prop=revisions&rvlimit=500&rvprop=ids|flags|user|userid|timestamp|size|sha1|parsedcomment&formatversion=2&format=json
Status: 200
There are 33 revisions.
The revisions are made by 25 unique users

```

Now please implement the parser which will check if the URL (Wikipedia API) is available, output the service status in console. If it is available, please parse the returned JSON objects and output how many unique users have made revisions to article “CNN”.