THE UNIVERSITY OF SYDNEY

School of Information Technologies

COMP5347: Web Application Development
Week 12 Tutorial: Restful Service

Learning Objectives

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

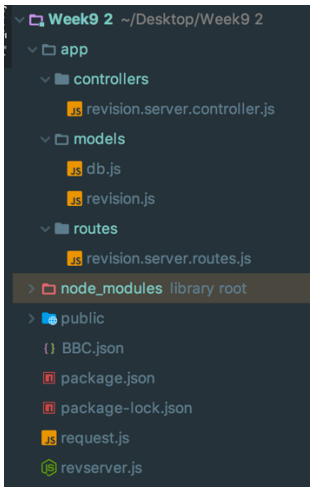
Part 1: Add request module

Task: Download the provided `week12-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 project, you need to run the following command in the terminal to install install mongoose 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 in Week 9 the `wikipedia` database with the `revisions` collections.

```
mongod --dbpath path
```

Run the code by right click the `/app/models/mongoose.revisions.js` on project explorer panel and run as `node.js application`.

Optional: line 3, the connection to mongodb is configured. If you give a different name to the Wikipedia data set when you import the data, the configuration might be different. In this case, you can replace the "dbname" in the following path with the name you created before. If you are not sure about the dbname, you may open the robomongo db browser and check it there.

'`mongodb://localhost/dbname`'

Right click `reverser.js` on the explorer panel and `Run as Node.js application`. Open the browser and go to `localhost:3000/revisions/revisions/BBC`. You will see something looks like the screenshot below.

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.cotroller.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.

In request.js, a request URL/endpoint is constructed with associated attributes. When you query this URL, JSON objects will be returns with the structure blow.

{"revid":757926565,"parentid":757331552,"user":"Lawarticles","userid":29835665,"timestamp":"2017-01-02T14:40:35Z"},

Now please implement the parse which will check if the URL above 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 “Australia”.