Notes RESTful Web Services

Application Programming Interface

The following document outlines the Notes Web Services interface implemented as part of the required Levels Beyond assessment project. It also contains information on how to launch the application, what tools were used in testing the interface and some brief design overview.

This document is included with the source code and application package. To launch the application you must have java 1.8 installed or Maven on your workstation and you must be located in the root directory where you extracted the package:

java -jar target/NotesRestApi-0.0.1-SNAPSHOT.jar
or
mvn spring-boot:run

You should now be able to connect to http://localhost:8080 and validate the interfaces I've implemented below. If you wise to open the server on a different port just update the src/main/resources/application.properties and change the server.port field to the port number that you prefer. Port number below 1000 require root access.
The source code is located under the following directory:
src/main/java/com/jboyd/notesApi

1. Development Overview

The application was written in JAVA 1.8 (https://www.java.com/en/download/) using the Spring Boot (https://spring.io/projects/spring-boot) framework which allows for rapid application/microservices development. I used maven (https://maven.apache.org/) as the build and deployment tool. Both curl and POSTMAN (https://www.getpostman.com/downloads/) were used to test the interfaces. I tested the application on both Ubuntu 16.04 and Ubuntu 18.04. This document was created using Libreoffice (https://www.libreoffice.org/). I used Eclipse (https://www.eclipse.org/) as my IDE. I documented the code using swagger (*https://swagger.io/*).

The main processing code can be found under the following directory:
src/jmain/ava/com/jboyd/notesApi
The web services controllers can be found under:
src/main/java/com/jboyd/notesApi/controllers
The integration test are under
src/test/java/com/jboyd/notesApi/
and can be run using the maven command:
mvn test
Logging goes to the following file:
logs/spring-boot-logger-log4j2.log

I prepopulated the data store with 7 entries.

There is also online documentation:
http://localhost:8080/api/swagger-ui.html#/

2. Web Services API Request Summary

The following list of tables provides the details on the interfaces that are supported. For the URI field, you must include "HTTP://localhost:8080/" and then the supported URI's reported below when using the interfaces.

Note: There is also documentation online, just use the following URI:
http://localhost:8080/api/swagger-ui.html#/

Table 1: GET api/notes

Description: Retrieve the list of note objects.

URI	***api/notes***
Method	**GET**
User Role Access	No restrictions
Version	0.0.1
Parameters	N/A
Request Header	Optional: Accept:application/json or application/xml Default: JSON
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/json or application/xml Default: JSON
Response Data	A list of notes

Example:

curl -i -X GET "http://localhost:8080/api/notes"

curl -i -X GET "http://localhost:8080/api/notes"
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 03 Apr 2019 18:14:52 GMT

[{"id":1,"body":"Its Sharon's birthday tomorrow"},{"id":2,"body":"You Mom left a message on the phone"},
{"id":3,"body":"Need to pick the kids up from school today"},{"id":4,"body":"The kids have music class Friday"},
{"id":5,"body":"Having lunch with Mary on Friday"},{"id":6,"body":"Need to call Sharon today"},{"id":7,"body":"Mom is leaving on vacation tomorrow"}]

Table 2: GET api/notes/{id}

Description: Retrieve the host whose id is given by the URI path template "id".

URI	*api/hosts/{id}*
Method	**GET**
User Role Access	No restrictions
Version	0.0.1
Parameters	N/A
Request Header	Optional: Accept:application/json or application/xml Default: JSON
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/json or application/xml Default: JSON
Response Data	A host object, or null if no object is found with the requested id

Example:

curl -i -X GET "http://localhost:8080/api/notes/5"
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 03 Apr 2019 18:24:27 GMT

{"id":5,"body":"Having lunch with Mary on Friday"}

Table 3: POST api/notes

Description: Create a new note resource.

URI	*anagramList*
Method	**POST**
User Role Access	No restrictions
Version	0.0.1
Parameters	N/A
Request Header	Content-Type:application/json or application/xml
Request Data	A note object
Response Codes	201
Response Header	Content-Type:application/json or application/xml Default: JSON
Response Data	The new note object

Example:

curl -i -X POST --header "content-type: application/json" -d '{ "body" : "Test note" }' http://localhost:8080/api/notes
HTTP/1.1 201
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 03 Apr 2019 18:36:46 GMT

{"id":8,"body":"Test note"}

Note: You must specify the content-type **‘--header "content-type: application/json"’**

Table 4: PUT api/notes/{id}

Description: Update the note object whose id is given by the URI path template “id”.

URI	***api/notes/{id}***
Method	**PUT**
User Role Access	No restrictions
Version	0.0.1
Parameters	N/A
Request Header	application/json or application/xml
Request Data	The note to modify. The note id must match the path template id.
Response Codes	200, 400, 404
Response Header	Content-Type:application/json or application/xml Default: JSON
Response Data	The update note

Example:

curl -i -X PUT --header "content-type: application/json" -d '{ "id":8, "body":"Update Test note" }' http://localhost:8080/api/notes/8
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 03 Apr 2019 19:00:19 GMT

{"id":8,"body":"Update Test note"}

Table 5: DELETE api/hosts/{id}

Description: Delete the note object whose id is given by the URI path template “id”.

URI	***api/notes/(id)***
Method	**DELETE**
User Role Access	No restrictions
Version	0.0.1
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	204, 404
Response Header	N/A
Response Data	Empty body.

Example:
curl -i -X DELETE http://localhost:8080/api/notes/8
HTTP/1.1 204
Content-Length: 0
Date: Wed, 03 Apr 2019 19:04:19 GMT