**URL Data retrieve App**

**Murshed** Dated: 11.06.2019

--------------------------------------------------

**Problem Analysis**

--------------------------------------------------

Develop an application that makes http (and https) requests to specified URLs and to report on certain properties of the responses it receives.

**Solution Design**

--------------------------------------------------

The application developed in JDK 1.8 using Java HttpURLConnection class and calling the method to fetch data on URL HTTP response header.

There are two classes UrlHttpRequest.java as helper class with method implemented and called them in main class UrlContentCheckApp.java

An array of response element from header data is created stautsArray and Json Object created for each URL input.

**Algorithm for Analysis and design**

------------------------------------------------------

1. Take user input as specified format – the list of URL
2. Split string and create an array
3. Loop through the array and invoke method for each URL
4. Return Json Object as result of each URL

Methods in helper class

1. getStatus()
2. Create instance of HttpURLConnection class
3. Open connection and timeout for 10 secs
4. Retrieve elements of HTTP Get request header
5. Return array with the elements of status code, content length and date-time
6. getJsonObject()
7. Get instance of Json Object
8. Invoke getStatus() method and copy return response elements in new array
9. Insert each Array elements to create Json Object
10. Print result as Json Object
11. Return Json Object useful for unit testing

The return Json Object getJsonObject and return Array from getStatus() does provide isolation and low coupling which enhance provision for unit testing.

**Coding and Testing**

--------------------------------------------------

I used approach fail-fast behaviour as a methodology of test driven development. The steps I followed

- Create helper class UrlHttpRequest with getStatus() and getJsonObject() methods.

- Create a class UrlContentCheckApp to implement get user input and invoke methods from helper classs

Before going further down to coding created JUnit test class UrlHttpRequestTest.java with first few test cases which obvious failed as unimplemented.

Moving to UrlHttpRequest class started coding getStatus() method until test cases pass.

Git version control used for commit and push in master branch. As this individual development I worked on Master node and push as order of application developed.

The project code available at –

https://github.com/MurshedCode/Check-URL-Api.git

**Test cases Assumptions**

--------------------------------------------------

Mainly two scenarios are considered. One for URL connection test data and other for testing Json Object created. Other random target assumption has been considered for reliability and fault tolerance issues.

**URL connection Check -**

testValidUrl() – invoking Java method to test whether url is valid

testGetContentOk() -

testDoesNotGetContentOk()

**Json Object check**

emptyJsonObject()

testNullJsonObject()

testNullJsonObjectDate()

testNullJsonObjectStatusCode()

testJsonObjectSimilar()

**Dependencies**

--------------------------------------------------

Junit5

java-json.jar

**Future Enhancement**

--------------------------------------------------

Four elements retrieved in the array for base of Json Object. This could be in any order or as more number of elements.

The input validation can be implemented for user choice of delimiter to convert. This could cover \t, \n, \| and other range of options. The application can take tab delimiter input at this stage. To change in any other format can update variable delimiter ="\t" in the UrlContentCheckApp class for instance new Line.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Refer to design architecture in analysis and design solution the application includes following files

UrlHttpRequest.java

UrlContentCheckApp.java

Test class

UrlHttpRequestTest.java

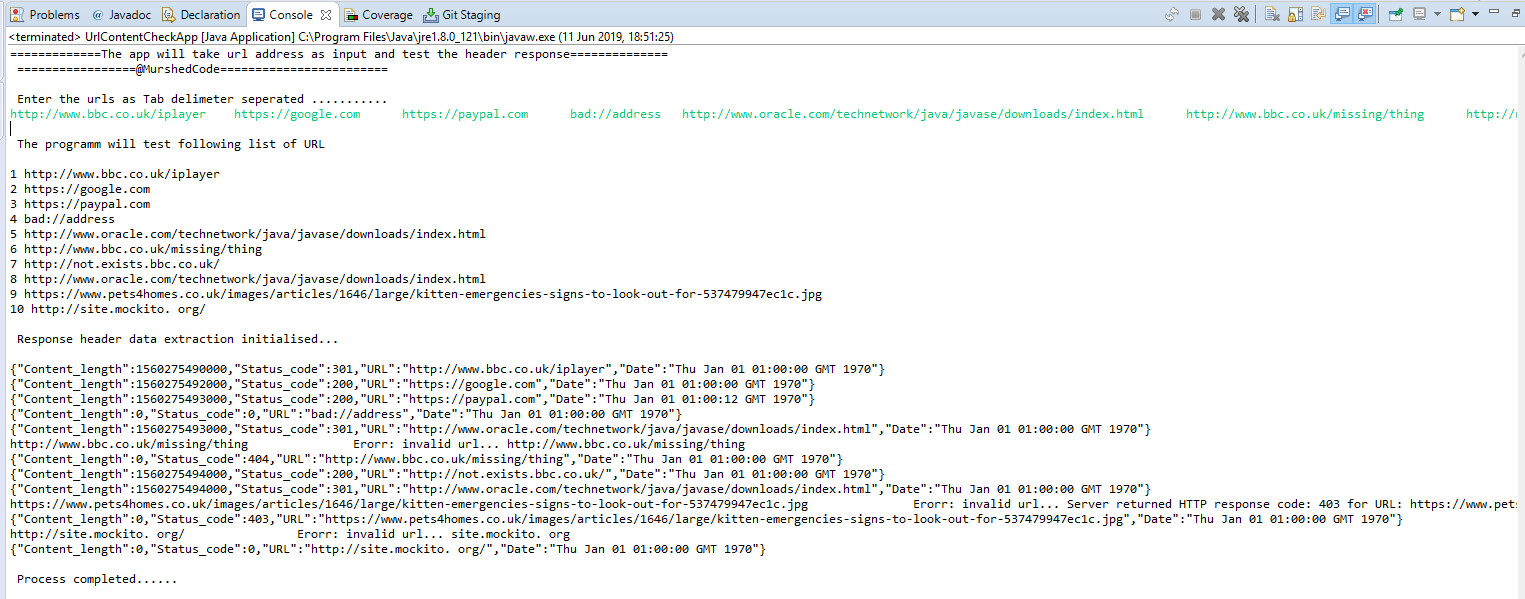
UrlContentCheckAppTest.java

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The class UrlContentCheckApp.java run as Java application and produce the output in console. Sample example of output:

Output at Eclipse Console

==========================================================



Output at command line running thorugh xxxxxx.Jar file

=========================================================

To dos

The app zipped as **Murshed\_Thakur.zip;** extract and import in your IDE and run RomanConverterApp.java class