SE 3XA3

REST Assured

October 27, 2017

Test Plan

Team 31

Dawson Myers	myersd1	400005616
Yang Liu	liuy136	400038517
Brandon Roberts	roberb1	400018117

Contents

List of Tables

List of Figures

1 Revision History

Date	Version	Notes
27-Oct-2017	0.0	Revision 0

Table 1: **Revision History**

2 Project Drivers

3 General Information

3.1 Purpose

The purpose of this document is to outline the testing, validation, verification procedures to be implemented for the reconstruction of the Sails Live Chrome app, named the REST Assured Test Client. Through testing, the REST Assured team aim to improve the products correctness and build confidence in the its functionality. The test plan helps provide proof that the project adheres to requirements specified in the Software Requirements Specification document. Types of testing will include structural, static and dynamic, functional and nonfunctional, manual and automated unit testing. Various testing tools will be used to achieve these tests. Fault testing will occur throughout the course of implementation of the project.

3.2 Scope

The scope of testing aims to cover the system fault conditions, These testing procedures may be complemented by design review and code review as a strategy to improve outcomes. The REST Assured team aims to test early and often to reduce faults with minimal expenditure of resources and to maximize correctness, quality, and reliability of software for users.

3.3 Naming Conventions and Terminology

3.4 Overview of Document

This document begins with a general overview of the plan, including sections on the software description, introduction of the test team and tools used for testing, and the Testing Schedule which will be followed. Next, detailed system test descriptions of functional and nonfunctional tests are presented, followed by tests for proof of concept. The document ends with the comparison to any existing implementation, and closing with unit testing plans.

Term	Definition
HTTP	Hypertext Transfer Protocol.
REST	Representational state transfer (REST) or RESTful web services is a way of providing interoperability between computer systems on the Internet.
JSON	JavaScript Object Notation. An open-standard file format that uses human-readable text to transmit data objects consisting of attributevalue pairs and array data types (or any other serializable value).
API	Application Program Interface. A document detailing the name of each function the client may call in their software and the purpose of those functions.
FR	Functional requirements that describes what the product will do.
User	A person who will be using the final product.
App	The application being designed; the system-to-be.

Table 2: **Table of Definitions**

4 Plan

4.1 Software Description

The REST Assured Test Client will provide software developers a tool in Web API building and testing. The application will provide tests endpoints and the capability to diagnose bugs in applications featuring RESTful interfaces.

4.2 Test Team

The REST Assured team members responsible for all testing procedures are Dawson Myers, Brandon Roberts, and Yang Liu. These responsibilities include test writing and execution for various types of testing outlined in this document.

4.3 Automated Testing Approach

Automated testing for the REST Assured Test Client will be done in Webstorm, which incorporates a wide array of test plugins which may be configured to benefit JavaScript debugging.

4.4 Testing Tools

The majority of the project code is JavaScript front-end code. The following testing tools will be used:

- Karma (Integration Tests)
- PhantomJS (UI Testing)

- Jasmine (Unit Testing)

4.5 Testing Schedule

See Gantt Chart at the following url: Team 31 Gant Project

Task	Team Member	Date
PoC Testing	Dawson Myers	November 27, 2017
FRT-UI-1	Brandon Roberts	November 12, 2017
FRT-UI-2	Brandon Roberts	November 12, 2017
FRT-UI-3	Brandon Roberts	November 12, 2017
FRT-UI-4	Yang Liu	November 12, 2017
FRT-UI-5	Yang Liu	November 12, 2017
FRT-UI-6	Yang Liu	November 12, 2017
FRT-UI-7	Dawson Myers	November 12, 2017
FRT-UI-8	Dawson Myers	November 12, 2017
FRT-UI-9	Dawson Myers	November 12, 2017
FRT-UI-10	Dawson Myers	November 12, 2017
FRT-P-1	Brandon Roberts	November 3, 2017
FRT-P-2	Yang Liu	November 3, 2017
FRT-CM-1	Yang Liu	November 21, 2017
NRT-P-1	Dawson Myers	November 21, 2017
NRT-P-2	Brandon Roberts	November 21, 2017

Table 3: **Testing Schedule**

5 System Test Description

The software will allow users to test their REST servers responses to GET/POST/PUT/DELETE requests. It will be implemented with common front end languages (HTML, javascript, css) and libraries (react, jQuery, bootstrap).

5.1 Tests for Functional Requirements

5.1.1 User Input

FRT-UI-1

Type	Manual
Initial State	Request form has input data, and response form has response information
Input	clear button clicked
Output	Request form and response form are cleared, leaving no characters in field
Procedure	The function clearing the request form and response form will run, the tester will manually verify if both forms have been cleared

FRT-UI-2

Type	Functional, Dynamic, Manual, Static etc.
Initial State	Input text fields empty
Input	clear button clicked
Output	Field remains cleared, no characters in field
Procedure	Manually perform test to verify if field has been cleared

FRT-UI-3

Type	Manual
Initial State	Input text fields cleared by clear button
Input	HTTP POST/GET/DELETE/PUT requests to test url
Output	HTTP request returns output fitting to request criteria
Procedure	Manually perform test to verify whether field clearing action will interfere with HTTP request functionalities

FRT-UI-4

Type	Manual
Initial State	The selected test stub is open
Input	The user clicks another test stub in the test selection menu
Output	The test stub viewer will update to display information about the newly selected test stub
Procedure	The test will manually be performed by a tester, and the program will pass the test if the wanted behaviour is reflected

FRT-UI-5

Type	Manual
Initial State	Test stub view is displayed
Input	HTTP POST/GET/DELETE/PUT requests to test url
Output	Test stub will change colour to corresponding request colour in the test selection menu
Procedure	The test will manually be performed by a tester, the functions corresponding to the HTTP requests will be run, we check the response and the program will pass the test if the desired behaviour is reflected

FRT-UI-6

Type	Manual
Initial State	Request form is awaiting input data
Input	User inputs request data into request form
Output	Program will format data for HTTP request with parameters, fit for browser entry
Procedure	The test will manually be performed by a tester, and the program will pass the test if the wanted behaviour is reflected

FRT-UI-7

Type	Manual
Initial State	A valid request entry has been entered in the request form as input data
Input	The save request entry button is clicked
Output	The saved request entry is added to the list of saved entries and appears in the saved entry selection window
Procedure	The save request entry function will be called for the input request entry, the test will manually verify that the input request is added to the saved list of requests and appears in the saved entry selection window

FRT-UI-8

Type	Manual
Initial State	Request form has been cleared of input data
Input	A previously saved request entry is selected, submit button is clicked
Output	Request form has been loaded with the selected previously saved request entry as input data
Procedure	The load saved request entry function will be called for the selected request entry, the test will manually verify that the request form has been populated with the selected request

FRT-UI-9

Type	Manual
Initial State	The program is open
Input	The user clicks the new test button
Output	A new test stub is created underneath the lowest test stub
Procedure	The test will manually be performed by a tester, the functions corresponding to the HTTP requests will be run, we check the response and the program will pass the test if the desired behaviour is reflected

FRT-UI-10

Type	Manual
Initial State	The program is openThe program is openThe program is open
Input	The user clicks and drags a test stub
Output	The test stub will follow the cursor users cursor until the let go by the user
Procedure	The test will manually be performed by a tester, the functions corresponding to the HTTP requests will be run, we check the response and the program will pass the test if the desired behaviour is reflected

5.1.2 Protocol Tests

FRT-PT-1

Type	Functional
Initial State	At main window
Input	Properly formatted JSON
Output	Should return true
Procedure	How test will be performed: REST query string validator function is called with a JSON request object

FRT-PT-2

Type	Functional
Initial State	At main window
Input	Improperly formatted JSON
Output	Should return false
Procedure	How test will be performed: REST query string validator function is called with a JSON request object

5.1.3 HTTP Communications

FRT-CM-1

Type	Functional
Initial State	At main window
Input	JSON request object
Output	JSON response object containing the correct set of data from the resource URL
Procedure	Test is run that will call the sendMsg function with a JSON request object. The function should return a JSON object with a data set from the server. The data will be validated to verify it is correct

5.2 Tests for Nonfunctional Requirements

5.2.1 Performance

NRT-P-1

Type	Functional
Initial State	At main window
Input	100,000 requests are enqueued
Output	JSON responses
Procedure	A test will add 100,000 request objects to the send message queue. The app should be able to process the responses without becoming unresponsive. The response text box should only store the previous 1000 rows of text

NRT-P-2

Type	Functional
Initial State	At main window
Input	JSON request for a very large data set
Output	JSON response
Procedure	A test will run that will make a request for a very large data set. The app should not become unresponsive while processing the response

6 Tests for Proof of Concept

6.1 User Input

PCT-UI-1

Type	Functional
Initial State	Main window waiting for request information
Input	User inputs request information
Output	The program unfolds the request information into a JSON object
Procedure	The test will manually be performed by a test member, and the program will pass the test if the wanted behaviour is reflected

PCT-UI-2	
Type	Functional
Initial State	Request form has input data, and response form has response information
Input	User clicks the clear button
Output	The Request form, and response form should be cleared of all information
Procedure	The test will manually be performed by a tester, and the program will pass the test if the wanted behaviour is reflected

7 Comparison to Existing Implementation

The existing project had very few test cases. Thus, the team has had to develop tests from scratch.

8 Unit Testing Plan

Jasmine will be used to test internal functions.

8.1 Unit testing of internal functions

Jasmine will also be used to test program output.

8.2 Unit testing of output files

N/A

References

None

9 Appendix

Additional information

9.1 Symbolic Parameters

Symbolic Parameters The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

Term	Definition
RESOURCE_ROOT_URL	https://jsonplaceholder.typicode.com
RESOURCE_POSTS	/posts
RESOURCE_COMMENTS	/comments

Table 4: **Table of Symbols**