

SE 3XA3  
REST Assured  
October 6, 2017

---

# Software Requirements Specification

---

Team 31

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

# Contents

<b>1</b>	<b>Revision History</b>	<b>1</b>
<b>2</b>	<b>Project Drivers</b>	<b>1</b>
2.1	The Purpose of the Project . . . . .	1
2.2	The Stakeholders . . . . .	1
2.2.1	The Client . . . . .	1
2.2.2	The Customers . . . . .	2
2.2.3	Other Stakeholders . . . . .	2
2.3	Mandated Constraints . . . . .	2
2.3.1	Solution Constraints . . . . .	2
2.3.2	Off-the-Shelf Software . . . . .	2
2.3.3	Schedule Constraints . . . . .	3
2.3.4	Budget Constraints . . . . .	3
2.3.5	Enterprise Constraints . . . . .	3
2.4	Naming Conventions and Terminology . . . . .	3
2.5	Relevant Facts and Assumptions . . . . .	3
<b>3</b>	<b>Functional Requirements</b>	<b>4</b>
3.1	The Scope of the Work and the Product . . . . .	4
3.1.1	The Context of the Work . . . . .	4
3.1.2	Work Partitioning . . . . .	4
3.1.3	Individual Product Use Cases . . . . .	5
3.2	Functional Requirements . . . . .	5
<b>4</b>	<b>Non-functional Requirements</b>	<b>6</b>
4.1	Look and Feel Requirements . . . . .	6
4.2	Usability and Humanity Requirements . . . . .	6
4.3	Performance Requirements . . . . .	6
4.4	Operational and Environmental Requirements . . . . .	7
4.5	Maintainability and Support Requirements . . . . .	7
4.6	Security Requirements . . . . .	7
4.7	Cultural Requirements . . . . .	7
4.8	Legal Requirements . . . . .	8
4.9	Health and Safety Requirements . . . . .	8
<b>5</b>	<b>Project Issues</b>	<b>8</b>
5.1	Open Issues . . . . .	8
5.2	Off-the-Shelf Solutions . . . . .	8
5.3	New Problems . . . . .	8
5.4	Tasks . . . . .	9
5.4.1	Project Planning . . . . .	9

5.5	Migration to the New Product . . . . .	9
5.6	Risks . . . . .	9
5.7	Costs . . . . .	9
5.8	User Documentation and Training . . . . .	10
5.9	Waiting Room . . . . .	10
5.10	Ideas for Solutions . . . . .	10
<b>6</b>	<b>Appendix</b>	<b>11</b>
6.1	Symbolic Parameters . . . . .	11

## List of Tables

1	<b>Revision History</b> . . . . .	1
2	Project Schedule . . . . .	9

## List of Figures

1	Context Diagram . . . . .	4
2	Send Message Use Case Diagram . . . . .	5

# 1 Revision History

Table 1: **Revision History**

Date	Developer(s)	Change	Revision
Oct 06, 2017	Dawson Myers, Yang Liu, Brandon Roberts	Revision 0	0
Nov 29, 2017	Dawson Myers, Yang Liu, Brandon Roberts	Revision 1 Proofread	1
Dec 6, 2017	Dawson Myers, Yang Liu, Brandon Roberts	Revision 1 Completion	1

This document describes the requirements for the REST client testing tool. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (?).

## 2 Project Drivers

### 2.1 The Purpose of the Project

Web services developed using web application programming interface (API) are omnipresent. Web APIs offers a powerful platform that enables HTTP services to a broad range of clients including browsers, mobiles, and desktops. They are created based on the representational state transfer (REST) framework that presents guidelines acting as a standard for Web APIs. It is critical to be able to test the API functionalities during its development stage. The reconstruction of the Sails Live Chrome app 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.

### 2.2 The Stakeholders

Stakeholders include the project team members developing the project who are learning about the software engineering project development process and the McMaster University Software Engineering 3XA3 course teaching staff who are invested in the educational value of the project for the development team. The product is not developed for commercial sale to customers, but rather as an open source utility tool free to developers who see a need for it. Therefore, the team itself is putting their own resources into the project.

#### 2.2.1 The Client

The client is Professor Asghar Bokhari who is teaching the McMaster University Software Engineering 3XA3 course. Since the project is intended to be open source and the core

project team sets the project direction, desired features and software requirements for the product, they may be considered as both the clients and the stakeholders.

### **2.2.2 The Customers**

The customers are developers who have a need for a REST client for testing their app. However, there app is a free utility, so they will not be paying customers.

### **2.2.3 Other Stakeholders**

Given the limited scope of the project, there are no other stakeholders beyond what has been listed above.

## **2.3 Mandated Constraints**

### **2.3.1 Solution Constraints**

The product shall operate on up-to-date versions of Windows, Mac, Linux operating systems with up-to-date versions of Chrome browsers installed that allow JavaScript to run. Rationale: Target client are users of up-to-date Windows, Mac, Linux operating systems with JavaScript client.

The product shall operate with a working and secure Internet connection with a download speed above 5 Mbps and an upload speed above 1 Mbps. The desired Internet connection speed is directly proportional to the quantity of data clients require for the testing of their specific web APIs (higher Internet connection speed is desired for applications with higher data traffic). Rationale: The product uses Internet connection to serve REST requests to required resources and to display request responses.

The product shall comply with Canadian federal privacy laws (Privacy Act and Personal Information Protection and Electronic Documents Act) and other relevant legal requirements. Rationale: The product should not breach legal requirements in the intended operating environment.

### **2.3.2 Off-the-Shelf Software**

The execution of product requires the following off-the-shelf software:

- Google Chrome web browser
- Client on which Chrome apps can be installed
- JavaScript

All of the above are freely available on the Internet.

### 2.3.3 Schedule Constraints

A working product must be completed to a state demonstrating full functionality for final demonstration by Nov 27, 2017. The product shall be completed for product release (along with all final documentation) by Dec 6, 2017.

### 2.3.4 Budget Constraints

The expected operation budget for the product is \$0, thus requirements for purchasing materials and other product needs are not applicable. Any unplanned purchases and costs during the development process, **such as web hosting cost**, are expected to be covered by the project team out of pocket.

### 2.3.5 Enterprise Constraints

The product is open-source and therefore freely accessible to all users with the technology and platform to run Chrome applications and JavaScript.

## 2.4 Naming Conventions and Terminology

- 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.
- Functional Requirements: 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.

## 2.5 Relevant Facts and Assumptions

It is assumed that the users will be software developers with a basic understanding of REST interfaces and the HTTP protocol.

## 3 Functional Requirements

### 3.1 The Scope of the Work and the Product

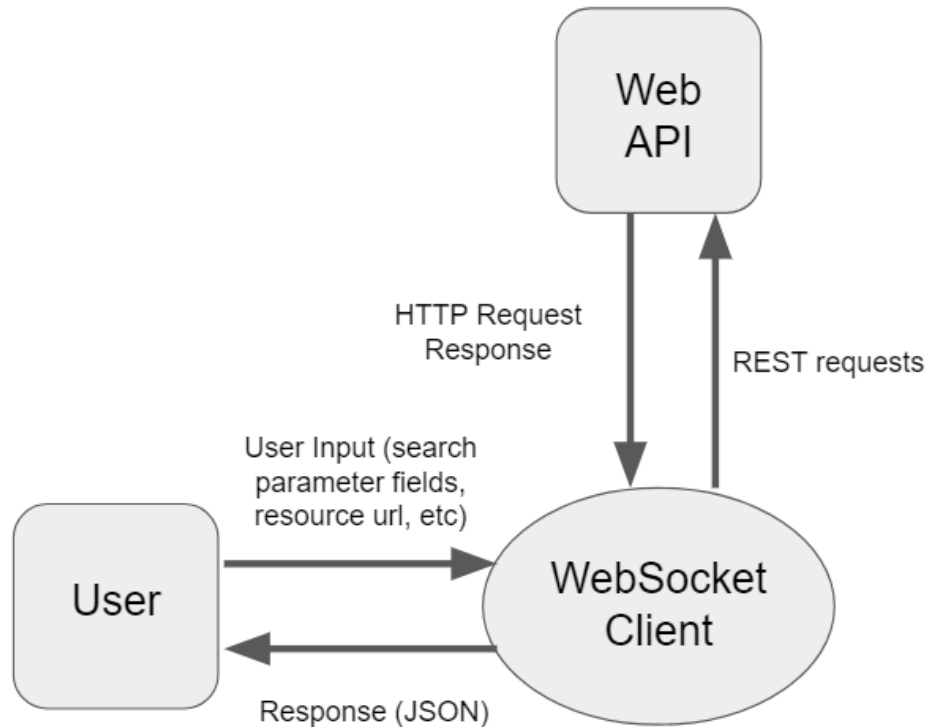


Figure 1: Context Diagram

#### 3.1.1 The Context of the Work

The recreation of the app is not limited to any particular platform since most devices are able to support the Chrome web browser and run JavaScript. The reproduction of the app will remove the limitation presented by Sails Live that users can interface with only the Sails server. Instead, the reproduced app aim to enable communication with interfaces featuring RESTful Web APIs not limited to Sails.

#### 3.1.2 Work Partitioning

Not relevant for this project.

### 3.1.3 Individual Product Use Cases

Figure 2 shows a use case for the app.

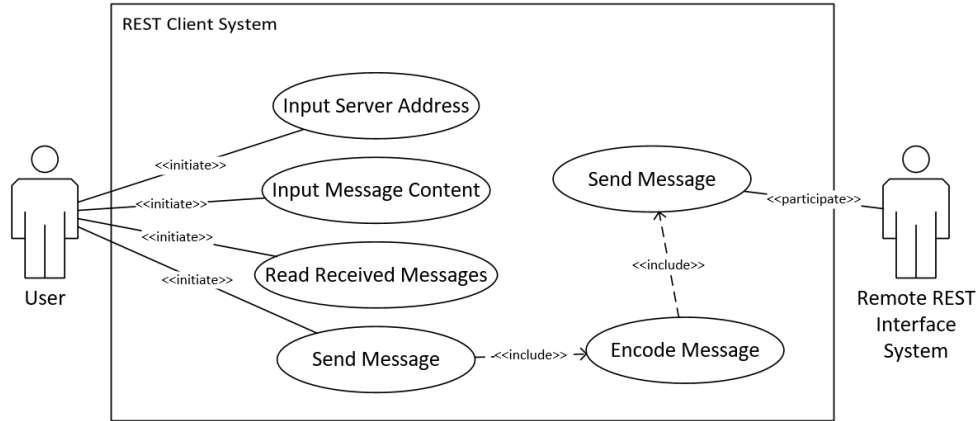


Figure 2: Send Message Use Case Diagram

## 3.2 Functional Requirements

**FR1**      *The app shall be able send and receive HTTP messages.*

*Rationale:*      In order to communicate information over the internet, the application will use the HTTP protocol.

*Fit Criterion:*      Messages should be able to be sent and received in less than one second.

*Priority:*      High

*Originator:*      Dawson Myers

*History:*      October 6, 2017

**FR2**      *Message format shall strictly follow the REST communication protocol.*

*Rationale:*      In order to simplify communication, the application will utilize the REST architectural style.

*Fit Criterion:*      Every message should follow protocols set forth by the formal REST architectural style.

*Priority:*      High

*Originator:*      Brandon Roberts

*History:*      October 6, 2017

**FR3**      *The app must provide a user interface that allows users to view the content of sent and received messages.*

*Rationale:*      In order to maintain high usability and feature visibility, and allows the user to easily translate goals into actions, the application will provide a user interface that is intuitive to navigate and to view the content of sent and received messages.

*Fit Criterion:*      More than 3/4 of test users should be able to navigate the web app without further instructions than labels presented on user interface.

*Priority:*      High

*Originator:*      Yang Liu

*History:*      October 6, 2017



## 4 Non-functional Requirements

### 4.1 Look and Feel Requirements

**NFR1**      *The app should have a simplistic and intuitive user interface.*

---

*Rationale:*      An application that is easy to use will make it more accessible to users. It will also increase productivity.

*Fit Criterion:*      At least 90% of users should be able to use the user interface to connect to a REST endpoint within 5 minutes of use.

*Priority:*      High

*Originator:*      Dawson Myers

*History:*      October 6, 2017

**NFR2**      *The app should be easy to install.*

---

*Rationale:*      95% of users should be able to locate and install the app via the Chrome App Store without encountering issues.

*Fit Criterion:*      Apps that are easy to install are more likely to be used.

*Priority:*      High

*Originator:*      Dawson Myers

*History:*      October 6, 2017

### 4.2 Usability and Humanity Requirements

**NFR3**      *In order to accommodate as many users as possible, the user should be able to adjust the font size for the app.*

---

*Rationale:*      An application that is accessible to people with poor eyesight or other disabilities will increase the applications potential user-base size.

*Fit Criterion:*      The user should not have to wait for more than 1 second for any operation to complete. If this is not possible due to network conditions, then there should be some kind of visible progress displayed to the user.

*Priority:*      Medium

*Originator:*      Dawson Myers

*History:*      October 6, 2017

### 4.3 Performance Requirements

**NFR4**      *The user interface should be responsive to user actions and not appear to be slow or hung up during normal operation.*

---

*Rationale:*      An application that responds quickly to the user leads to an overall more positive reception.

*Fit Criterion:*      The user should not have to wait for more than 1 second for any operation to complete. If this is not possible due to network conditions, then there should be some kind of visible progress displayed to the user.

*Priority:*      Medium

*Originator:*      Dawson Myers

*History:*      October 6, 2017

## 4.4 Operational and Environmental Requirements

**NFR5**      *The app should be able to run on any system that Google Chrome can run on.*

---

*Rationale:*      Google Chrome is offered on almost every operating system. Running the application through Google Chrome will maximize the apps potential.

*Fit Criterion:*      The app should be compatible with Windows, Linux, and Mac OS.

*Priority:*      Medium

*Originator:*      Dawson Myers

*History:*      October 6, 2017

## 4.5 Maintainability and Support Requirements

**NFR6**      *The app should be able to be updated on users machines in a simple, unintrusive and automated fashion.*

---

*Rationale:*      Keeping the application up-to-date in a simple, unintrusive and automated fashion is convenient and desirable for the end user.

*Fit Criterion:*      The app should utilize the updating mechanism that is used by the Chrome App Store.

*Priority:*      Medium

*Originator:*      Dawson Myers

*History:*      October 6, 2017

## 4.6 Security Requirements

**NFR7**      *The app should not, in any way, access private data on users computers.*

---

*Rationale:*      Accessing a users private information without express consent is a violation of the law. Therefore, the application will never access private information this without consent.

*Fit Criterion:*      Users must first give consent before any data is accessed by the app.

*Priority:*      Medium

*Originator:*      Dawson Myers

*History:*      October 6, 2017

**NFR8**      *The app should not make users computers vulnerable to any cyber threats.*

---

*Rationale:*      Protecting the users machine from external threats is important, and is mandated by the law in Canada.

*Fit Criterion:*      The app must not allow any form of remote access or subvert the cyber defence mechanisms on the host computer.

*Priority:*      High

*Originator:*      Dawson Myers

*History:*      October 6, 2017

## 4.7 Cultural Requirements

**NFR9**      *The app should not be offensive to users.*

---

*Rationale:*      Offensive material could lower productivity, or turn away potential users.

*Fit Criterion:*      The app should not be offensive to at least 85% of potential users. The application will launch without any material perceived offensive by team members.

*Priority:*      High

*Originator:*      Dawson Myers

*History:*      October 6, 2017

<b>NFR10</b>	<i>The app should be understandable to users that understand english, regardless of culture.</i>
<i>Rationale:</i>	Lack of cultural images and references makes it easier for users to understand the uses of the application.
<i>Fit Criterion:</i>	The app should not contain any cultural images or references.
<i>Priority:</i>	Medium
<i>Originator:</i>	Dawson Myers
<i>History:</i>	October 6, 2017

## 4.8 Legal Requirements

The application must not take personal information from the user without their express consent, in accordance with relevant Canadian and US laws. The app must not infringe on any copyright, trademarks, or intellectual properties.

## 4.9 Health and Safety Requirements

Health and safety are issues that should be considered for every engineering project. The product does not interact with users physically and thus do not impose physical health concerns to the users. Safety precautions should be taken to protect the privacy of users and safeguard any personal information. The product requirements and constraints taken into consideration that the product must abide by all Canadian federal privacy laws to achieve this standard.

# 5 Project Issues

## 5.1 Open Issues

There are currently no open issues.

## 5.2 Off-the-Shelf Solutions

Some off-the-shelf solutions that may be utilized for the application include JavaScript frameworks such as Angular, which is a structural framework for dynamic web apps that can be used for data binding and dependency injection. The cross-platform JavaScript library jQuery may be used to simplify the client-side scripting of HTML in application design. JavaScript libraries such as React may be used for building interactive user interfaces that efficiently update and render the desired components.

## 5.3 New Problems

Data storage is currently being discussed and will likely have an issue opened for it soon. The issue involves the format for storing configuration data on host machines.

## 5.4 Tasks

The tasks for the project is set forth by the McMaster University Software Engineering 3XA3 course deliverables outline.

### 5.4.1 Project Planning

The project will follow the schedule in shown in table 2.

Task Item	Revision	Timeline
Proof of Concept Demonstration	0	Oct 16, 2017
JavaScript Design	0	Oct 20, 2017
JavaScript Implementation	0	Oct 25, 2017
Test Plan	0	Oct 27, 2017
Design Document	0	Nov 10, 2017
Revision 0 Demonstration	0	Nov 13, 2017
Final Demonstration	1	Nov 27, 2017
Final Documentation	1	Dec 6, 2017

Table 2: Project Schedule

## 5.5 Migration to the New Product

Not applicable to this project.

## 5.6 Risks

One risk factor involved with the project is that automated testing may be difficult to implement due to variation in request responses to applications featuring live outputs (such as Twitter API). The project has been through a preliminary round of evaluation by the teaching staff of the course to ensure the planned scope has maximum potential for project success. However, in case the scope of the project is too large to complete within the designated project time line, there may be need to narrow the scope of the project to allow project completion.

## 5.7 Costs

The product is projected to incur no monetary cost as long as services such as source code hosting and version control systems are free of cost to the project team. The main cost is the time of the project team members who develop and maintain the product.

## 5.8 User Documentation and Training

It is important for users to learn how to use the application. Simple annotated screen shots will be provided to explain how to use the basic functions of the application.

## 5.9 Waiting Room

Features that are considered for future releases include:

- Socket connection support for remote applications and interfaces
- Real-time payload validation (JSON validation)
- Request management (edit, remove, sort or inspect created requests)
- Event registration (on which the connected socket should listen to)
- Create, manage and inspect listeners
- Observe realtime notifications
- Collapsible response inspector
- Listener queue (list and save all incoming events with date and time information)
- Multiple connections (select the socket you need for each request and listener)
- Connection history
- Automated API tests with configuration
- Import/Export of custom settings

## 5.10 Ideas for Solutions

~~Stated in previous sections.~~ The fulfillment of the functional and non-functional requirements can be accomplished through the development of a web application using tools mentioned in off-the-shelf solutions as well as through iterative test-driven development (according to the fit-criterion of each requirement).

## 6 Appendix

### 6.1 Symbolic Parameters

Symbolic parameters are yet to be determined.