

STANDARD REST API FOR EMAIL

Kalana Guniyangoda

(118209x)



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Dissertation submitted in partial fulfillment of the requirements for the
degree Master of Science

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

June 2015

DECLARATION

“I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:  University of Moratuwa, Sri Lanka Date:
Name: Kalana Guniyangoda www.lib.mrt.ac.lk
Electronic Theses & Dissertations

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the supervisor:

Date:

Name of the supervisor: Prof. Gihan Dias

ABSTRACT

Email has long been a most popular mode of electronic communication. Initially, email communication was between multi-user hosts using the SMTP protocol, and later on, with the popularity of client-server communication, protocols such as POP, IMAP and Submit were developed for connecting e-mail clients and servers. Today, the most popular method of e-mail access is via a web browser. However, there is still a lack of standard protocol defined for e-mail access via web browsers. All the current web-mail systems use proprietary communication between web interfaces and the backend server. Therefore, each web-mail system can only be accessed with its own web interface and vice versa. Therefore, it is opportune to develop a standard protocol for email servers and browser-based email clients harnessed with HTML5 capabilities to communicate over the HTTP protocol.

Representational State Transfer (REST) is a popular architectural style to implement applications using the HTTP protocol and offers many features such as scalability and loose coupling. This would be beneficial in implementing browser-based email clients and would make it possible to create an open standardised HTTP based protocol similar to SMTP.

In this dissertation, we analyse the major REST and non-REST HTTP-based e-mail protocols and APIs, starting from Paul Prescod's initial proposal, as well as other email protocols such as IMAP, and identify the set of features required of an http-based e-mail protocol. We then define a standard API for this purpose, combing the strong features of current systems and protocols. The REST API introduced in this dissertation provides the needed functionality of an e-mail system, including authentication, sending emails, reading emails and managing emails & attachments. Furthermore, we specify messaging formats, error codes and notification mechanisms for the system. We have also developed a server-side implementation which supports the API.

We have run the e-mail system under three scenarios, and show that it has acceptable functionality and performance.

ACKNOWLEDGEMENTS

I would like to dedicate my sincere thanks to my supervisor Prof. Gihan Dias for his dedicated support for the success of this research. This would not have become a success without your support from the initial stage of the research.

I would like to thank the entire academic and the non-academic staff of the Department of Computer Science and Engineering for their kindness extended to me in every aspect.

Last but not least, I thank my family and all my friends who supported me for the success of this piece of work. Your support was so precious.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES.....	viii
LIST OF TABLES	ix
LIST OF ABBREVIATIONS.....	x
LIST OF APPENDICES.....	xi
1. INTRODUCTION	1
1.1. Background.....	1
1.2. Objectives.....	2
2. LITERATURE SURVEY	3
2.1. Email.....	3
2.2. The history and evolution.....	3
2.3. Email Protocols	4
2.3.1. SMTP	4
2.3.2. POP and IMAP.....	5
2.4. Messaging Formats.....	6
2.4.1. Internet Message Format.....	6
2.4.2. Multipurpose Internet Mail Extensions.....	7
2.5. Web Services	7
2.6. Service Oriented Architecture (SOA)	8
2.7. Representational state Transfer (REST).....	9
2.7.1. Application of REST.....	11
2.7.2. Use of HTTP request methods.....	12

2.7.3.	HATEOAS	13
2.8.	REST vs. SOAP.....	14
2.9.	XML	15
2.10.	JSON.....	16
2.11.	JSON vs. XML	16
2.12.	Authentication mechanisms.....	18
2.12.1.	HTTP Authentication	18
2.12.2.	Query Based Authentication.....	19
2.12.3.	OAuth 2.0 Authorization Framework.....	19
2.13.	REST based email systems.....	20
2.13.1.	“Reinventing Email using REST”	20
2.13.2.	HTTP Access to Email Stores	21
2.13.3.	Restful interface for database based email server	23
2.13.4.	RESTMAIL by Marcin Bazydlo	25
3.	STUDY OF EMAIL APIs.....	26
3.1.	Gmail REST API.....	26
3.2.	Outlook Mail REST API.....	29
3.3.	Zimbra REST API	31
3.4.	Sendinc API.....	32
3.5.	Postmark REST API.....	34
3.6.	Email Yak REST API.....	36
3.7.	Context.IO Email REST API	37
3.8.	Mailgun REST API	38
3.9.	PostageApp API	39
3.10.	Yahoo! Mail Web Service	40
3.11.	Summary of Commercial API	41
3.12.	Comparison of HTTP methods uses in APIs	42
4.	COMPARATIVE ANALYSIS OF API FUNCTIONS.....	43
4.1.	Common functions of a generic email system	43
4.1.1.	Login into email system.....	43
4.1.2.	Listing email directories available in the account.....	44

4.1.3.	Listing mailbox content.....	44
4.1.4.	Renaming mailboxes.....	45
4.1.5.	Deleting mailboxes.....	46
4.1.6.	Display mail headers.....	46
4.1.7.	Mail retrieval.....	47
4.1.8.	Deleting Email messages.....	47
4.1.9.	Retrieval of attachments.....	48
4.1.10.	Email flag handling.....	48
4.1.11.	Copy/Move directories/ emails within directories.....	49
4.1.12.	Email Searching and filtering.....	49
4.1.13.	Email Sending.....	50
4.2.	Summary of Functions.....	51
5.	THE SYSTEM DESIGN.....	53
5.1.	Architecture.....	53
5.2.	HTTP Methods.....	55
5.3.	Data exchange language.....	55
5.3.1.	Hypermedia Format.....	56
5.4.	Resources.....	57
5.4.1.	Base URL.....	58
5.4.2.	Mail account.....	59
5.4.3.	Mail directory.....	59
5.4.4.	Mail.....	61
5.4.5.	Attachments.....	63
5.5.	Functionality.....	63
5.5.1.	Login to mail system.....	64
5.5.2.	Getting a list of mail directory.....	64
5.5.3.	Creating new mail directory.....	65
5.5.4.	Rename a mail directory.....	65
5.5.5.	Delete a mail directory.....	66
5.5.6.	Searching mails within a directory.....	67
5.5.7.	Listing emails in a directory.....	67

5.5.8.	Displaying email	68
5.5.9.	Retrieving email attachments	68
5.5.10.	Posting/Creating email	69
5.5.11.	Deleting email	70
5.5.12.	Flag manipulation.....	70
5.5.13.	Email sending.....	70
5.5.14.	Moving/Copying email.....	71
5.6.	Summary of Design.....	72
6.	IMPLEMENTATION.....	73
6.1.	Architecture	73
6.2.	Development environment	73
6.3.	Library usage	74
6.3.1.	PHP: IMAP	74
6.3.2.	Swift Mailer	75
6.4.	Apache Configurations.....	75
6.5.	API Configurations.....	76
6.6.	User agent/ Client.....	76
6.7.	Testing of the API	78
6.7.1.	Correctness of functionality	79
6.7.2.	Time consumption for each functionality	79
6.7.3.	Observations.....	81
7.	CONCLUSION & FUTURE WORK.....	82
	REFERENCES.....	83
	APPENDIX A: SOURCE CODE	86

LIST OF FIGURES

Figure 2-1: Mail Sending Process	5
Figure 5-1: Fully REST based email system	53
Figure 5-2: Hybrid REST mail system.....	54
Figure 5-3: Resources	58
Figure 6-1: Proposed Architecture	73
Figure 6-2: Google Chrome – REST client extension	77
Figure 6-3: CURL command line tool	77
Figure 6-4: Test Scenario 03	78
Figure 6-5 Test Scenario 02	78
Figure 6-6: Test Scenario 01	79



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF TABLES

Table 2-1: HTTP request methods 12

Table 2-2: REST vs. SOAP..... 14

Table 2-3: JSON vs. XML performance 18

Table 3-1: Mailgun customized error codes..... 39

Table 3-2: Summary of Commercial API 41

Table 3-3: Summary of HTTP method use in API 42

Table 4-1 : Functionality Analysis of Vendor specific APIs 51

Table 5-1: Proposed HTTP request methods 55

Table 5-2: Parameter list for mail resource..... 62

Table 6-1: List of PHP:IMAP functions used 74

Table 6-2: Correctness of Functions 79

Table 6-3: Performance analysis 80



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF ABBREVIATIONS

API	Application Programme Interface
HATEOAS	Hypermedia As The Engine Of Application State
HTTP	Hypertext Transfer protocol
HTTPS	HTTP over TLS
IANA	Internet Assigned Numbers Authority
IMAP	Internet Message Access Protocol
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
POP	Post Office Protocol
REST	Representational State Transfer
SMTP	Simple mail transfer protocol
TLS	Transport Layer Security
XML	EXtensible markup Language



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF APPENDICES

Appendix	Description	Page
Appendix - A	Source Code	115



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk