УДК 004.055

Разработка веб-службы для доступа к электронной почте на основе двухфакторной аутентификации

М. А. Мясников

Национальный исследовательский университет «МИЭТ»
e-mail: maksim.m00@mail.ru

Development of a web service for email access based on two-factor authentication

M.A. Myasnikov

National Research University of Electronic Technology,
Moscow, Russia,
e-mail: maksim.m00@mail.ru

В статье рассмотрен подход к созданию веб-службы, для повышение скорости разработки веб-приложений, включающих работу с различными почтовыми серверами и почтовыми ящиками. Рассматриваются вопросы выбора интерфейса доступа API, протокола доступа к электронным почтовым серверам и применения технологии многофакторной аутентификации.

Ключевые слова: почтовый сервер, двухфакторная аутентификация, RESTful-службы.

The article describes an approach to creating a web service to increase the speed of developing web applications that include working with various mail message transfer agents and mailboxes. The issues of choosing an API access interface, an access protocol to message transfer agent and the use of multi-factor authentication technology are considered.

Keywords: message transfer agent, two-factor authentication, RESTful-service.

В современном мире сложно представить такого человека, который бы не имел своего электронного почтового ящика. Более того, люди имеют в своем распоряжении по нескольку электронных адресов и не

обязательно на одном почтовом сервере. Обход и проверка всех почтовых ящиков на наличие новых писем, при их большом количестве превращается в довольно-таки сложную и затратную по времени задачу. Для решения этой проблемы хотелось бы иметь под рукой службу, которая соберет все письма и предоставит к ним доступ.

Конечно, идея создания единого сборщика электронных писем с различных адресов не является новой. Известно, что существуют разные почтовые клиенты, позволяющие управлять несколькими почтовыми аккаунтами. Однако их существенным недостатком является ограниченность и отсутствие функционала в приложение, будь то веб-интерфейс, десктопное или мобильное приложение, а также ограниченная возможность подстроить функционал под свои нужды. Помимо этого, не все десктопные/мобильные приложения являются кроссплатформенными.

В связи с этим является актуальным создание веб-службы (не зависящей от платформы, на которой она будет вызываться) с открытым интерфейсом, предоставляющим доступ к широкому спектру действий с электронными письмами различных адресов и серверов.

Для решения проблемы проектирования такой службы необходимо решить несколько вопросов:

- Какой интерфейс доступа к функционалу веб-службы использовать?
- Какой протокол использовать для работы с почтой?
- Как защитить и обезопасить данные пользователей веб-службы?

Веб-служба, также веб-сервис (Web service), – ресурс сети, предоставляющий информационное наполнение и (или) функциональные возможности, к которым можно обратиться дистанционно через стандартизированные протоколы и программные интерфейсы.

Основными способами взаимодействия с веб-службой являются протоколы SOAP, XML-RPC, а также соглашение REST.

Рассмотрим основные их различия в таблице 1.

Таблица 1 – Сравнение SOAP, XML-RPC и REST

Характеристика	SOAP	XML-RPC	REST
1	2	3	4
Как организован	Конвертная структура сообщений	Локальный вызов процедуры	Соответствие архитектурным ограничениям

Продолжение таблицы 1

1	2	3	4
			(единство интерфейса; отсутствие состояния; кэширование; клиент-серверный подход; многоуровневая система; расширяемость функционала)
Формат	XML	XML	XML, JSON, HTML, текст
Изучение и работа	Сложно	Легко	Легко
Сообщество	Небольшое	Большое	Большое
Использование	Платежные шлюзы; управление CRM-решениями финансовых и телекоммуникационных сервисов;	Командное и ориентированное на действия API; Высокопроизводительное взаимодействие большой микросервисной системы	API-интерфейсы управления; Простые приложения, управляемые ресурсами

Наиболее оптимальным и гибким способом является построение RESTful-службы, с поддержкой различных форматов ответа, по желанию пользователя.

Далее определимся с протоколом доступа получения почты с сервера: POP3 или IMAP, для чего сравним их основные возможности.

Для этого рассмотрим таблицу 2.

Таблица 2 – Сравнение протоколов IMAP и POP3

Характеристика	IMAP	POP3
1	2	3
Полное название	Internet Messaging Access Protocol – Протокол доступа к Интернет сообщениям	Post Office Protocol – Протокол почтового отделения
Место хранения писем	На сервере	На компьютере пользователя
Синхронизация	Возможность синхронизации на нескольких устройствах	Нет синхронизации на нескольких устройствах
Доступ к письмам с вложениями	Возможность скачать тело письма, без загрузки вложений	Письма скачиваются целиком, с вложениями
Возможность потери данных	Письма будут утеряны только при поломке сервера, копии могут быть сохранены на локальных устройствах	Письма будут потеряны на локальном устройстве, без возможности восстановления (удаляются с сервера при получении)
Необходимость постоянного интернет-соединения	Необходимо постоянное интернет-соединение для чтения и написания писем. Возможность просмотреть только текст письма при медленном интернет-соединении	Необходимость интернет-соединения только для единовременного скачивания или отправки письма. Невозможность увидеть только текст письма, необходимость загрузки всех вложений (даже при

Продолжение таблицы 2

1	2	3
		слабом интернет-соединении)
Кому подойдет	Подойдет для пользователей, которым необходим доступ к почте с нескольких устройств	Подойдет для пользователей, работающих с почтой с одного устройства

Как можно видеть, каждый из протоколов имеет свои преимущества и недостатки. Но так как проектируется универсальная служба, а также есть вероятность, что не все почтовые серверы поддерживают работу с одним или другим протоколом, реализуем работу с обоими протоколами и дадим возможность выбора для пользователей.

Для защиты писем пользователей веб-службы, от взлома аккаунта, на который служба будет собирать письма с электронных адресов, усложним идентификацию пользователя аутентификационными данными двух типов, иначе говоря, будем использовать двухфакторную аутентификацию. Для этого пользователь должен иметь два из трех типов данных идентификации:

- То, что ему известно;
- То, что он имеет;
- То, что ему присуще.

То, что присуще пользователю, – это биометрические данные; но в рамках проектирования данной службы затрагивать их не будем.

То, что известно пользователю, – это его логин и пароль, которые он хранит в секрете от остальных.

Под тем, что пользователь имеет, изначально понимался токен, как некоторое компактное устройство. Сейчас же в роли токена чаще выступает телефон. Таким образом, тем, что имеет пользователь может быть:

- Присылаемый пользователю код на другой адрес электронной почты или на телефон с помощью SMS;
- Отдельное устройство с дисплеем, отображающим код либо подключаемое к компьютеру через системы USB или Bluetooth;
- Приложение на телефоне, генерирующее код.

Получать код на электронную почту, в рамках построения веб-службы, предоставляющей доступ к электронной почте – не уместно; код

пришедший по SMS можно подсмотреть даже без разблокирования телефона; устройства-токены не представляется возможным распространять на широкую аудиторию. Поэтому выбор будет отдан приложению на телефоне, которое будет генерировать код.

Чтобы пользователь мог получить доступ к работе со службой он должен будет указывать свои логин, пароль, а также код, генерируемый приложением с определенной периодичностью.

Можно использовать уже готовые приложения, как например Google Authenticator или Microsoft Authenticator, либо же создать такое самому. Возникает новая задача: узнать, предоставляют ли разработчики приложений возможность использовать их алгоритмы на сторонних проектах, либо изучить алгоритмы TOTP (Time-based One-time Password, создаёт код по известному серверу и приложению ключу а также текущему времени) и HOTP (HMAC-based One-time Password, основан на счетчике, изменяющемся при каждом запросе кода. HMAC - hash-based message authentication code, код на основе хеш-функции)

Таким образом, были рассмотрены и приняты решения по двум из трех вопросов поставленных для решения проблемы проектирования веб-службы для доступа к электронной почте:

- Какой интерфейс доступа к функционалу веб-службы использовать? – Проектировать RESTful-службу с различными форматами возвращаемых данных
- Какой протокол использовать для работы с почтой? – Использовать комбинацию сервисов IMAP и POP3 для различных ситуаций.
- На последний вопрос «Как защитить и обезопасить данные пользователей веб-службы?» дан частичный ответ – использовать двухфакторную аутентификацию с генерацией подтверждающего кода в мобильном приложении. Однако возник новый вопрос о реализации взаимодействия мобильного приложения с проектируемой веб-службой.

Библиографический список:

1. Олифер Виктор, Олифер Наталья. Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание. — СПб.: Питер, 2020. — 1008 с.

2. Андресс Джейсон. Защита данных. От авторизации до аудита. — СПб.: Питер, 2021. — 272 с.