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*

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 большем 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 нова. Существуют разные почтовые клиенты, позволяющие управлять несколькими почтовыми аккаунтами. Однако их существенным недостатком является «вшитость» функционала в приложение, будь то веб-интерфейс, десктопное или мобильное приложение и ограниченная кастомизируемость. Помимо этого, не все десктопные/мобильные приложения являются кроссплатформенными. Преимущественно будет создать платформонезависимую веб-службу с открытым интерфейсом, предоставляющий 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | SOAP | XML-RPC | REST |
| Как организован | Конвертная структура сообщений | Локальный вызов процедуры | Соответствие архитектурным ограничениям (единство интерфейса, отсутствие состояния, кэширование, клиент-серверность, многоуровневая система, расширяемость функционала) |
| Формат | XML | XML | XML, JSON, HTML, простой текст |
| Сложность изучения и работы | Сложно | Легко | Легко |
| Сообщество | Небольшое | Большое | Большое |
| Использование | Платежные шлюзы, управление CRM-решениями финансовых и телекоммуникационных сервисов, поддержка legacy-систем | Командное и ориентированное на действия API; Высокопроизводительное взаимодействие большой микро-сервисной системы | API-интерфейсы управления;  Простые приложения, управляемые ресурсами |

Табл. 1. Сравнение SOAP, XML-RPC и REST

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.

|  |  |  |
| --- | --- | --- |
| Характеристика | IMAP | POP3 |
| 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 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 |

Табл. 2. Сравнение протоколов IMAP и POP3

Как видим, каждый из протоколов в чем-то лучше, а в чем-то хуже. Но так как мы проектируем универсальную службу, а также есть вероятность что не все почтовые сервера поддерживают работу с одним или другим протоколом, реализуем работу с обоими протоколами и дадим возможность выбора для пользователей.

Для защиты же 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 и в рамках проектирования данной службы мы касаться 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; токенные устройства 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 написать такое самому. Возникает лишь цель узнать, предоставляют ли разработчики приложений возможность использовать их алгоритмы сторонним проектам, либо изучить алгоритмы OATH TOTP (Time-based One-time Password, создающим код по известному серверу и приложению ключу а также текущему времени) и OATH HOTP (HMAC-based One-time Password, основанный на счетчике изменяюшимся при каждом запросе кода. Прим. HMAC - hash-based message authentication code)

Т.о. 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 же вопрос «Как защитить и обезопасить данные пользователей веб-службы?» дан частичный ответ – использовать двухфакторную аутентификацию с генерацией подтверждающего кода в мобильном приложении. Однако пока еще остается открытым вопрос о реализации взаимодействия мобильного приложения с проектируемой веб-службой.

Список литературы: