Salesforce Email Integration Security Guide

Salesforce, Spring '24

@salesforcedocs

Last updated: November 14, 2023

CONTENTS

SECURITY GUIDE OVERVIEW

The Salesforce integrations with Outlook and Gmail help sales reps manage their sales more efficiently, regardless of where they choose to complete their work. The integrations with Outlook and Gmail are available at no cost with Sales Cloud.

Note: Starting in late 2023, existing Inbox services and data are migrating to Hyperforce. Hyperforce is Salesforce cloud-native infrastructure architecture, built for the public cloud. Before the migration, some Inbox services and data are stored in Salesforce-managed data centers in Germany or the United States, and hosted on Amazon Web Services (AWS) behind a Virtual Private Cloud (VPC). Post-migration, the Inbox services and data are built on Hyperforce and stored on new AWS public cloud infrastructure within the same region.

This document covers technical and security guidelines for:

- The Outlook and Gmail integrations.
- Desktop and mobile solutions when an Inbox license is present and users are assigned an Inbox permission. An Inbox license is available with Sales Cloud Einstein, Sales Engagement, and as a standalone license.

The addition of an Inbox license provides:

- More features in the Outlook and Gmail integrations to increase sales reps' productivity while they're working in Outlook and Gmail.
- Access to select Inbox features in email from Lightning Experience.
- Access to Inbox mobile apps.

Complete information, including setup steps, considerations, and details about the features are available in Salesforce inbox in Salesforce help.

Salesforce offers other features and solutions to integrate email accounts with Salesforce that complement the Outlook and Gmail integrations and Inbox features. For example, set up Einstein Activity Capture or Lightning Sync to sync contacts and calendar events with Salesforce. And, set up automated email and event logging with Einstein Activity Capture.

For security considerations, see the Einstein Activity Capture Security Guide and the Lightning Sync Design and Security Guide.

Note: An Inbox license includes Einstein Activity Capture. However, you can enable Inbox with or without the Einstein Activity Capture feature. You can also enable Einstein Activity Capture with or without Inbox.

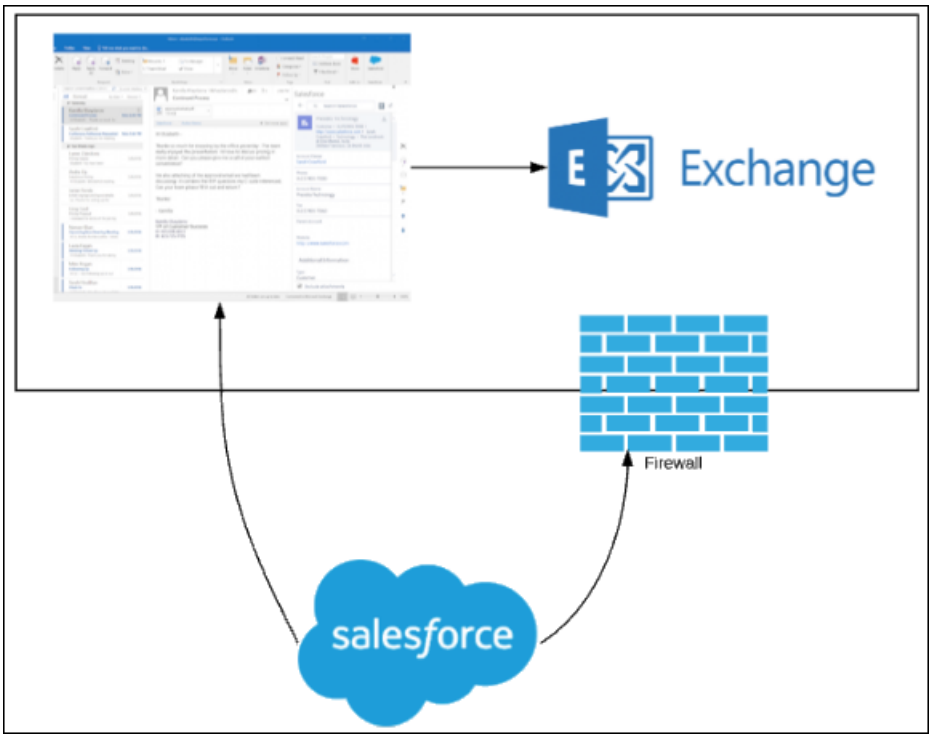
OUTLOOK INTEGRATION

Make good choices when granting access to your Exchange server for the Outlook integration.

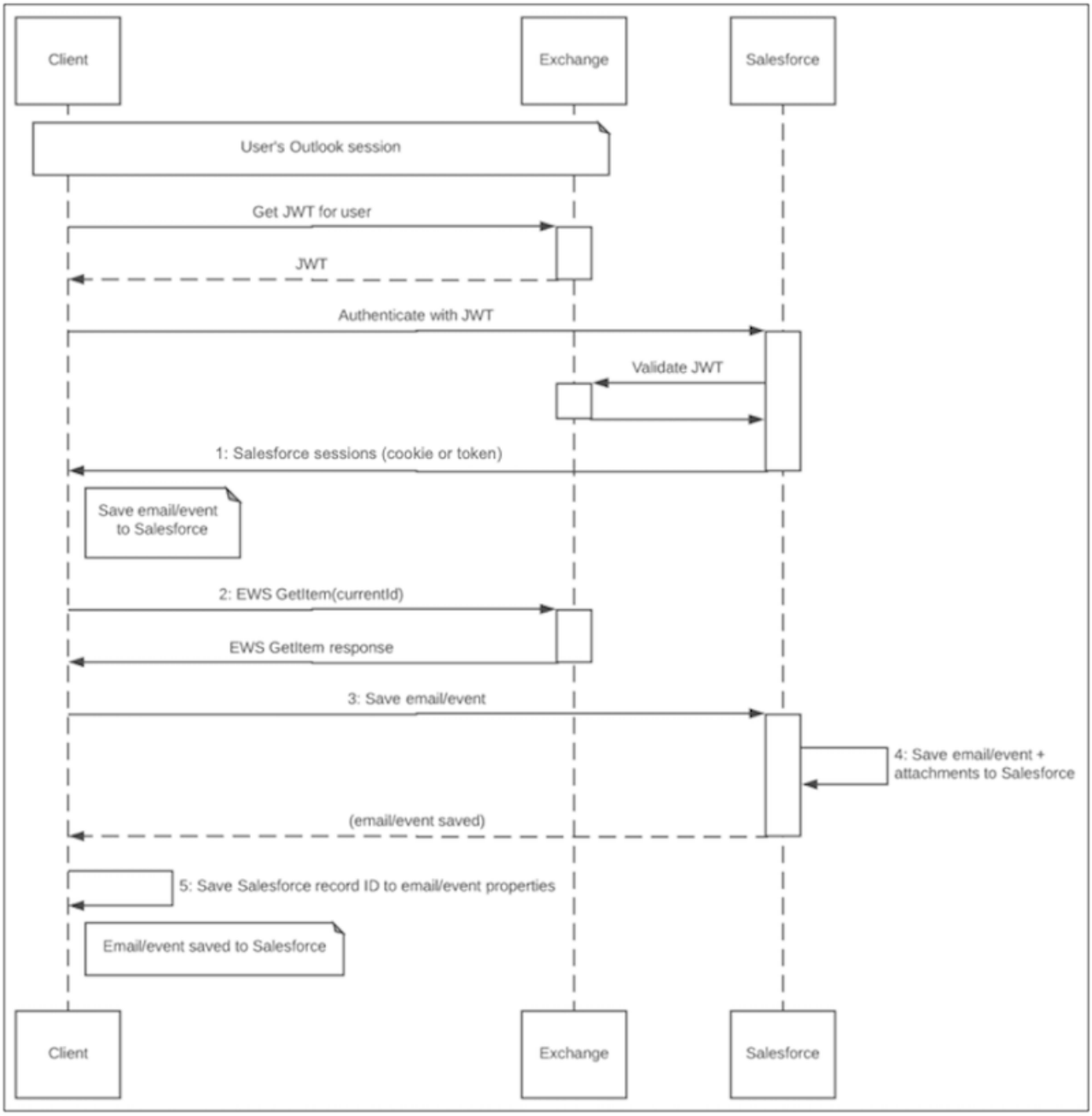
Setting up the Outlook integration requires access to your Exchange server. How you choose to set up that access depends on the versions of Outlook you use, your internal security policies, and the features that sales reps need within the integration.

The Outlook integration add-in is built on the Microsoft Office Add-In Framework. To log emails from Outlook to Salesforce (among other end-user actions) within that framework, Salesforce is required to make calls to the Exchange server.

In a typical Exchange on-premises setup, a firewall blocks access from the internet.



The Outlook integration taps into the Exchange API and places Exchange Web Services (EWS) calls from Salesforce application servers. Historically, the add-in calls were placed with an Exchange-provided JSON Web Token (JWT) at the URL provided by Exchange itself, via EWS. The JWT calls required an exposed EWS endpoint and still does for older versions of Exchange and Outlook.



With recent Microsoft enhancements in modern versions of Outlook and Exchange, the historic EWS server calls can be client calls in the Office.js API that Outlook provides. With the correct versions of Outlook and Exchange, there's no need to expose an EWS endpoint to power almost all the features in the Outlook integration. However, a local EWS connection is still required between Outlook and Exchange and the Exchange Metadata URL must still be publicly exposed.

If Exchange and Outlook run JavaScript API v1.8 or later, there's no need to expose an EWS endpoint to power the standard Outlook integration features. However, a local EWS connection is still required between Outlook and Exchange, and the Exchange Metadata URL must still be publicly exposed. This change in setup is available on a rolling basis to existing customers starting in Summer '21. For details about timing and eligibility, contact your Salesforce account representative.

The latest builds of Exchange Online run JavaScript API v1.8 or later. To determine if your Outlook client runs the JavaScript API v1.8 or later, see Outlook JavaScript API requirement sets in the Microsoft documentation.

Important: Features available with an Inbox license, such as insert availability and send later, require access to the Exchange server, regardless of the Outlook or Exchange API version. If you have an Inbox license, review Outlook Integration with a Public EWS Endpoint on page 6 and Outlook and Gmail Integrations with an Inbox License on page 11.

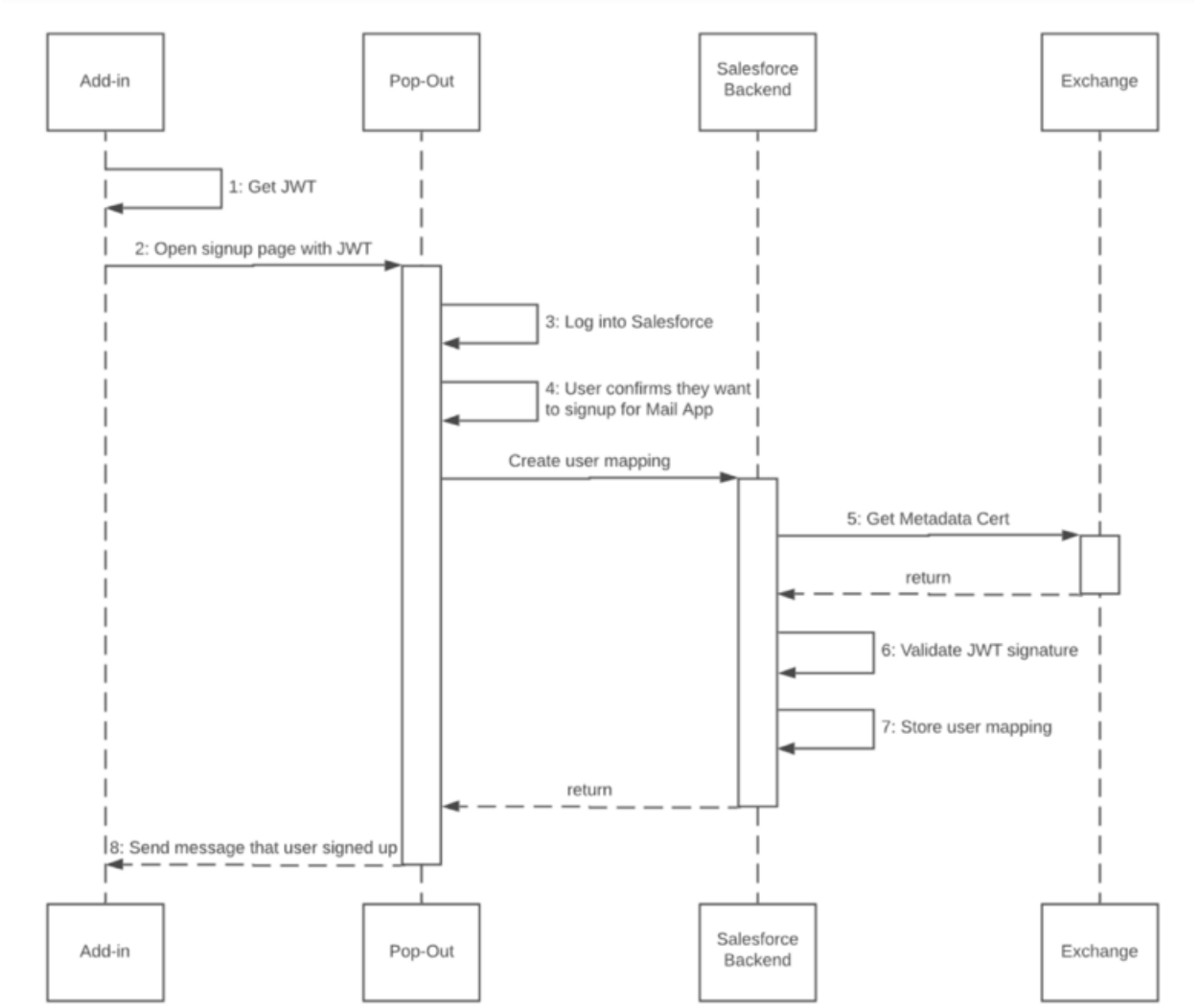
If your Exchange server or Outlook versions support JavaScript AP versions 1.4 through 1.7, you can still choose to set up Exchange without public EWS. However, users lose access to the following features:

- Logging attachments directly from Outlook. Users can add attachments to logged emails in Salesforce, seeing "Logged to Salesforce" indications on emails and events that have been logged to Salesforce.
- Inbox productivity features.

First-Time User Authentication Login Flow

Salesforce connects to Exchange to authenticate a user via the metadata URL and is a separate consideration from EWS.

This diagram details the flow for how the Exchange mail is mapped to the corresponding Salesforce users the first time they load the Outlook integration add-in. This flow applies to all versions of Outlook and Exchange, regardless of the JavaScript API version.



1. The Outlook add-in retrieves an identity token with a simple JavaScript method:

   ```
   Office.context.mailbox.getUserIdentityTokenAsync(callback, userContext);
   ```

   The JavaScript method requests an Exchange user identity token (a JSON Web Token or JWT) from the Exchange server. The add-in opens the sign-up page in a window hosted on Salesforce.

2. The user authenticates with their Salesforce credentials.
3. Salesforce prompts the user to connect their Exchange account (specified in the identity token) with the authenticated Salesforce user.
4. The user clicks the prompt, confirming they want to sign in.
5. Salesforce serves then validates the Exchange token contents and fetches the public certificate of the metadata URL. Salesforce expects the EWS endpoint to have a valid certificate. See Salesforce Help for information about supported SSL certificates.
6. Salesforce validates the identity token signature by accessing the public signing key from the authentication metadata document on the Exchange server.

   When the Exchange server initially provides the JSON Token to the add-in, it specifies the following:

   - An Exchange Metadata Endpoint URL inside the payload part of the token itself

- The Salesforce add-in

The add-in sends a request to the defined metadata URL to validate the signature. The Exchange metadata URL must be publicly accessible for validation of the user's identity token.

To learn more about validating a token, see Microsoft documentation.

7. The Exchange to Salesforce user mapping is then stored within the user's Salesforce org data.

Outlook Integration with a Public EWS Endpoint

The Outlook integration add-in uses authenticated calls in several scenarios.

- Outlook versions are running JavaScript API 1.7 or earlier. Check which version of the API your Outlook application runs in Outlook JavaScript API requirement sets.
- You've added an Inbox license, which enables features including insert availability, sent later, text shortcuts, and email tracking. These features require access to the Exchange server. Also review Outlook and Gmail Integrations with an Inbox License on page 11 in this guide. That section includes security and implementation considerations beyond what is discussed in this section.

Important: Without the public EWS endpoint in these scenarios, integration users can't log attachments from the integration or use any Inbox productivity features.

Configuration Requirements

Ensure your Outlook integration is configured correctly.

Configuration Requirements for Outlook on the Web

Certain settings apply when your reps use the Salesforce integration in Outlook on the web.

Logging Emails with Attachments to Salesforce Flow

Manually log selected Outlook email messages and attachments to Salesforce.

APIs Used

Learn about API and Exchange Web Services (EWS) calls.

Exchange Web Services (EWS)

Exchange Web Services requests must be formatted correctly.

EWS APIs Used

Learn more about Exchange Web Services calls.

Configuration Requirements

Ensure your Outlook integration is configured correctly.

Configuring the Outlook integration requires the public exposure of URLs.

- Exchange metadata URL that permits unauthenticated HTTP access. See the First-Time User Authentication Login Flow on page 4
- Exchange Web Service URL

Configuration Requirements for Outlook on the Web

Certain settings apply when your reps use the Salesforce integration in Outlook on the web.

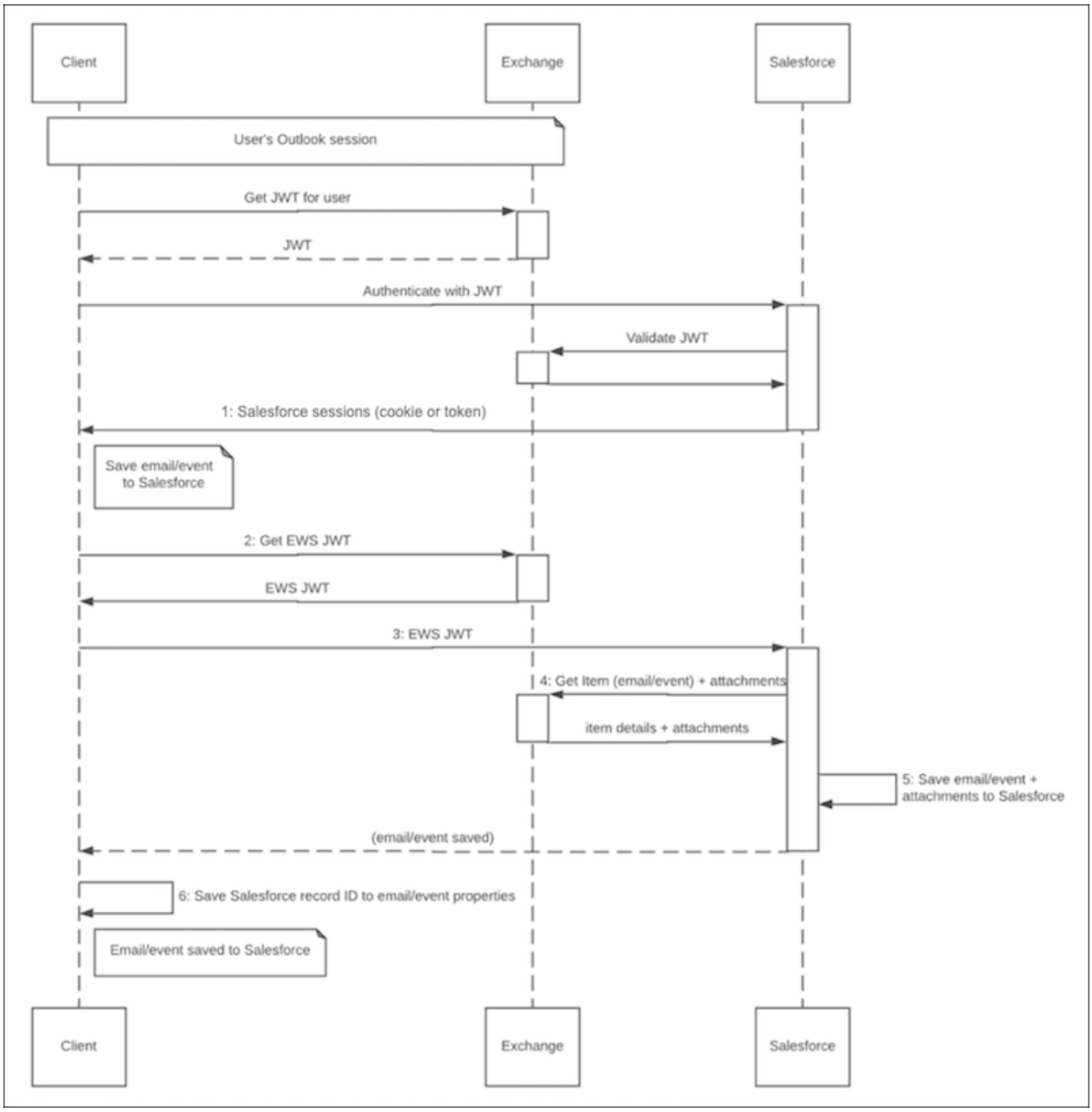
Because Salesforce makes outgoing calls to Exchange endpoints, each endpoint URL must have a valid SSL certificate. Learn more about all the SSL certificates supported by Salesforce.

If your reps use Outlook on the web (also known as the Outlook Web App (OWA)), specify any custom OWA URLs, such as non-Office 365 URLs, in the Outlook integration settings in Salesforce setup. Custom URLs don't require public exposure because only the client browser needs access to Outlook on the web.

Logging Emails with Attachments to Salesforce Flow

Manually log selected Outlook email messages and attachments to Salesforce.

From the Outlook integration, users can manually log a selected Outlook email message and its attachments to Salesforce. The add-in uses the following flow to complete the logging:



1. Authenticates with Salesforce (see Login flow) for details.
2. Makes an authenticated call to Exchange Web Services (EWS) via the API provided to Outlook add-ins. Salesforce servers are now allowed to fetch the current email or event to be logged. See Microsoft Office API documentation.
3. Performs the EWS operations EWS GetItem + GetAttachment(s) for the current email or event and its attachments.
4. Saves the email or event and the attachments to Salesforce and associates both to the selected Salesforce records.
5. Modifies the email or event in Exchange to include the Salesforce record ID in the extended properties of the Exchange object.

APIs Used

Learn about API and Exchange Web Services (EWS) calls.

We make client-side API calls via Office.js and server EWS calls, limited to GetItem and GetAttachment operations. The EWS calls that we make are initiated from the client side and from the Salesforce app servers. A user action triggers these calls in the context of a particular email or event. The calls coming from the Salesforce app servers to your EWS URL come from the published IP address ranges.

The Outlook integration specifies ReadWriteMailbox so that it can read the email or event and its attachments. The Write access is to write the Salesforce task or event ID back to the Exchange record via an EWS call placed through the Office.js API. See the Office.js documentation for details about the configuration requirements for making this EWS call.

Exchange Web Services (EWS)

Exchange Web Services requests must be formatted correctly.

The EWS request contains:

- HTTP headers
  - Authorization: Bearer token (from Office.js getCallbackTokenAsync)
  - User-Agent: ExchangeServicesClient/0.0.0.0
- SOAP request body XML

EWS APIs Used

Learn more about Exchange Web Services calls.

We make the following calls via EWS to get emails or events and their attachments. We also write the Salesforce record ID to the properties of the Exchange item. Click the links for Microsoft documentation about the specific calls.

- GetItem (client side and server side) to get and set AdditionalProperties and the content of the current email message when saving to Salesforce records
- GetAttachment (server side) to retrieve the attachments from Exchange and add to Salesforce records (associated with the Salesforce email message representation)
- UpdateItem (client side)
- GetFolder (client side) to get the drafts folder
- CreateItem (client side), which we use to create a draft message

"Client side" refers to calls made via the Office.js API makeEwsRequestAsync. "Server side" refers to calls made from Salesforce app servers to EWS endpoint. For these server-side calls, we use a five-minute token from getCallbackTokenAsync.

GMAIL INTEGRATION

This section covers login authentication and the authenticated calls that the features in the Gmail integration Chrome extension use.

If your email integration includes Inbox, also review the Outlook and Gmail Integrations with an Inbox License section of this guide.

Configuration Requirements

Make sure that your system meets the requirements before you integrate with Gmail.

To set up the integration with Gmail, review the system requirements and other considerations in Set Up the Integration with Gmailin Salesforce Help.

Authentication

Learn about the authentication methods Salesforce uses with Gmail integration.

OAuth 2.0

Salesforce uses the OAuth 2.0 protocol to connect to a user's Google accounts. The Salesforce server obtains and stores an OAuth refresh and access token for making requests to Google. This token is a single-user token that provides access to that user's Gmail account. The Chrome extension doesn't use this token directly. It's stored within the connected Salesforce org and treated as customer data.

Authentication Providers

The Gmail integration uses Authentication Providers, a Salesforce platform feature, to store and manage the Google access tokens. Authentication Providers allow Apex to retrieve the access token and to refresh it. To learn more, see Authentication Providers in Salesforce Help

Keep Gmail and Salesforce Connected

To stay logged in with Google, enable the Keep Gmail and Salesforce Connected preference, available on the Gmail Integration and Sync page in Salesforce Setup. This preference allows users to obtain a Salesforce session based on their Google identity. The Salesforce session follows the expiration time and other rules, such as allowable IP range, as set within Salesforce. When the Salesforce session expires, users can establish a new session based on their Google identity. This setup requires users' browsers to be logged into their Google accounts. When the preference is disabled, users log in the same way that they log in to Salesforce desktop. The same admin-controlled session rules apply. When the Salesforce session expires, users are required to log in again.

OUTLOOK AND GMAIL INTEGRATIONS WITH AN INBOX LICENSE

The addition of an Inbox license unlocks more features to increase sales reps' productivity within the Outlook and Gmail integrations. It also provides sales reps access to more features in email from Lightning Experience and provides access to Inbox mobile apps.

Important: The content of this chapter only applies if you have Salesforce Inbox. If you don't have an Inbox license, or none of your users are assigned an Inbox permission, you can skip this chapter.

This chapter details connectivity, data storage, and data retention when an Inbox license is present and users are assigned an Inbox permission.

Org Provisioning

Learn about where Inbox data is stored.

Note: Hyperforce is Salesforce infrastructure architecture, built for use with public cloud providers, such as Amazon Web Services (AWS). Hyperforce is composed of code rather than hardware, so that the Salesforce platform and applications can be delivered rapidly and reliably to locations worldwide. It provides Salesforce applications with compliance, security, privacy, agility, and scalability, and gives customers more choice and control over data residency.

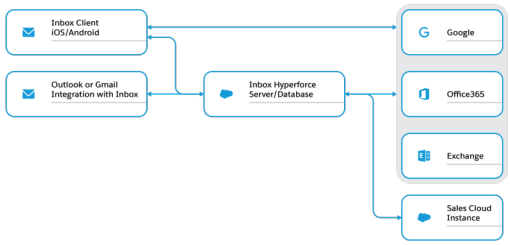
When Inbox is enabled in a Salesforce org, a corresponding org is created on Salesforce's Hyperforce servers. When users have permission to use Inbox and connect their email mailbox to Salesforce, their email mailbox is connected to Hyperforce. This connection prompts the Salesforce Hyperforce servers to make network calls to Google, Microsoft Exchange, or Office 365. Within the Hyperforce data centers, our application uses keys and IDs to ensure that we serve the relevant data to the relevant customers.

For information about the security and architecture of the Einstein Platform that Inbox uses, see the Einstein Platform Trust and compliance documentation.

Network Connections

Learn about the network calls between Salesforce and connected applications.

Inbox mobile apps and desktop clients make network calls to the Salesforce Hyperforce servers. Then, the servers make direct network calls to Microsoft Exchange, Office 365, and Google.



Outlook and Gmail integrations with an Inbox License to Salesforce Hyperforce Servers
An HTTPS TLS 1.2 connection with AES-128 cipher. This connection is used for login and for performing Inbox-specific tasks.

Salesforce Hyperforce Servers to Google
An HTTPS TLS 1.2 connection with AES-256 cipher. This connection uses the Gmail API protocol with OAuth 2.0 authentication.

Salesforce Hyperforce Servers to Office 365 (Exchange Online)
An HTTPS TLS 1.2 connection with AES-256 cipher. This connection uses the EWS protocol with OAuth 2.0 authentication.

Salesforce Hyperforce Servers to On-Premises Exchange (2019, 2016, and 2013)
An HTTPS TLS 1.2 connection (the Exchange server decides the TLS version and cipher). This connection uses the EWS protocol with username and password basic authentication. If an IP or VPN restricts the EWS endpoint, add the following addresses to the allowed list of addresses.

Important: Before migration to Hyperforce, some data is stored on AWS servers and databases behind a Virtual Private Cloud (VPC). To ensure uninterrupted access of your Inbox services and data, add both sets of IP addresses to your allowlist.

	If Your Salesforce Instance Is Outside of Europe	If Your Salesforce Instance Is in Europe
Hyperforce	• 44.242.15.232 • 44.236.183.129 • 100.21.196.196 • 54.200.249.136 • 44.228.8.56 • 35.165.2.200	• 18.158.21.76 • 3.76.67.243 • 18.158.241.92 • 3.76.75.66 • 52.57.103.81 • 3.72.121.255
Pre-Hyperforce	• 54.200.130.205 • 54.218.59.121 • 34.210.91.105 • 34.210.91.103 • 44.224.62.36 • 52.35.129.120 • 54.71.145.62 • 35.166.120.106 • 44.224.71.98 • 52.35.232.62 • 54.68.117.123 • 52.26.6.102 • 35.163.187.73 • 52.36.92.175 • 34.210.91.106 • 34.210.91.104 • 35.166.17.212 • 34.216.184.173 • 34.210.91.108 • 34.210.91.107	• 52.59.28.245 • 52.28.30.206 • 52.57.191.228 • 18.194.116.65 • 52.57.191.229 • 18.184.19.133 • 52.57.191.226 • 52.57.191.224 • 52.57.191.227 • 18.197.233.154 • 18.196.51.181 • 3.124.138.13 • 3.124.208.146 • 3.124.224.62 • 3.124.238.55

See this knowledge article for information about Salesforce IP addresses that aren't specific to EWS or to Inbox.

If you have restrictions on Exchange outbound connections, allow outbound access as indicated in the table. Then, when new emails and events arrive in Exchange, push notifications are sent to Salesforce. To ensure uninterrupted access of your Einstein Activity Capture services and data, add both sets of webhook endpoints to your org.

	If Your Salesforce Instance is Outside of Europe	If Your Salesforce Instance is in Europe
Hyperforce	apiq-ews-webhook-c01.sfdc-lywfpd.svc.sfdcfc.net	apiq-ews-webhook-c01.sfdc-yzvdd4.svc.sfdcfc.net
Pre-Hyperforce	ews-webhook-us1-prod.salesforceiq.com	ews-webhook-eu1-prod.salesforceiq.com

iOS and Android Inbox App to Google
An HTTPS TLS 1.2 connection with AES-256 cipher. This connection uses the Gmail API protocol with OAuth 2.0 authentication.

iOS and Android Inbox App to Office 365 (Exchange Online)
An HTTPS TLS 1.2 connection with AES-256 cipher. This connection uses the Exchange ActiveSync (EAS) protocol or the Office 365 API protocol, each with OAuth 2.0 authentication.

iOS and Android Inbox App to On-Premises Exchange (2019, 2016, and 2013)
An HTTPS TLS 1.2 connection. The Exchange server decides the TLS version and cipher. This connection uses the Exchange ActiveSync (EAS) protocol with username and password basic authentication.

Salesforce, Hyperforce, and Amazon Web Services (AWS) Servers Storage

When an Inbox license is enabled, servers and databases are hosted on Hyperforce and built on public cloud infrastructure, such as Amazon Web Services (AWS).

Important: An Inbox license comes with Einstein Activity Capture. However, you can enable Inbox with or without enabling the Einstein Activity Capture feature. Because both features use the same data, the data is stored on Hyperforce, regardless of Einstein Activity Capture being on.

Review details about what data is captured and stored, and how the data is used.

What Hyperforce Captures and Stores	Additional Details	How Data is Used
Calendar events	Calendar events include all event data that comes from users' connected Microsoft or Google accounts. They don't include event attachments.	Einstein Activity Capture uses the data to display events in the activity timeline and the Salesforce calendar. Inbox uses the date for the Insert Availability and Recommended Connections features.
Contact details	The details include contact data from what's displayed in the Contact Profile screen from Gmail, Exchange, or Sales Cloud.	Contact data is used by other Salesforce features, such as Einstein Email Insights.
Email accounts	The information includes details about users' connected Microsoft or Google accounts, including email address, server, and domain.	Einstein Activity Capture and Inbox use the data to connect users' email accounts to Salesforce.
Email attachments	The metadata for email attachments is included. For Einstein Activity Capture, the attachments themselves aren't stored or shown on the activity timeline. For Inbox, the Send Later feature stores the attachments until the email is sent. During Inbox's email send action, attachments can be Email Attachments dynamically fetched from the Google or	Einstein Activity Capture doesn't currently use the attachment metadata. Inbox uses the attachments and metadata for the Send Later feature.

What Hyperforce Captures and Stores	Additional Details	How Data is Used
	Exchange server by passing the email message ID.	
Email headers and metadata	The email messages are from users' connected Microsoft or Google accounts. The email elements that are stored include: Subject, From, To, CC, and sent date.	Einstein Activity Capture uses the data to add emails to the activity timeline of related Salesforce records. Email Insights, available with Inbox and Einstein Activity Capture, uses the data to create classifications. Recommended Connections, which is available with Inbox and Einstein Activity Capture, uses the data to generate suggestions.
Email HTML bodies		Einstein Activity Capture uses the data to display emails in Salesforce. The data is also used to generate email insights.
Passwords and OAuth tokens	The OAuth refresh and access tokens are used to connect users' Google or Microsoft accounts to Salesforce. When users connect their account to Salesforce with OAuth 2.0, we don't store users' passwords. Therefore, if users change their email password after connecting their account to Salesforce, they don't have to reauthenticate against Google or Microsoft. For users that use on-premises Exchange email accounts that use password authentication, we store users' passwords.	
Salesforce records	The records also include metadata, such as permissions, fields, and page layouts, for records such as contacts, leads, and opportunities. Inbox stores metadata for records for up to 24 hours. Einstein Activity Capture stores metadata for records until you delete the data in Salesforce. When you delete the record data in Salesforce, it's also removed from Hyperforce servers.	Inbox mobile apps use the data to improve performance when looking up records related to an email or event. To associate emails with related Salesforce records, Einstein Activity Capture copies email addresses from contact and lead records and stores them on Hyperforce servers.
User settings	The user settings include the user's personal settings from Inbox or Einstein Activity Capture.	

Hyperforce Data Retention

Inbox stores some data using the Salesforce Hyperforce infrastructure.

When a new email account is connected, the Salesforce Hyperforce servers download 6 months or 180 days into the past. From that point on, the servers use notification subscriptions from Gmail, Exchange, or Office 365 to trigger downloading new email messages. By default, email messages are retained for two years before being deleted. You can configure the retention period from 30 days up to 5 years.

Encryption Key Management

By default, Salesforce encrypts all at-rest customer data on Hyperforce at the infrastructure layer, using capabilities supplied by the cloud provider.

All data is encrypted at-rest and in-transit, and Salesforce owns the keys for encrypting data. Hyperforce keys aren't org-specific.

The email metadata and calendar data in the database is encrypted using disk-level encryption.

Data Storage for Inbox Mobile Apps

Review what information Inbox mobile apps store.

What	Details
Inbox User Settings	Personal user settings are stored in an SQLite database on the device and in the app preference. • NSUserDefaults on iOS • SharePreferences in Android The user settings stored include: • Salesforce settings (prompt to log email) • A user's set work hours • Email Settings, including swipe direction, number of lines to show in a message, organize by thread, and badge count type • Calendar settings, including declined events and number of days to display .
Email Messages	Inbox mobile apps store recent email messages, including the email body. Email messages are stored in an SQLite database. On iOS, the database is encrypted using SQLCipher, which uses AES-256 encryption. On Android, the database isn't encrypted.
Calendar Events	Inbox mobile apps store calendar events in an SQLite database. On iOS, the database is encrypted using SQLCipher, which uses AES-256 encryption. On Android, the calendar events are stored in the default shared calendar provider storage. See Android developer documentation for information on the calendar.
Passwords and OAuth Tokens	To fetch email and calendar events directly from Google, Office 365, and Exchange, Inbox mobile apps store OAuth access tokens from Google and Office 365. They also store passwords for Exchange. These tokens or passwords are stored in the iOS keychain or Android Account Manager, which are the default encrypted areas for the respective devices.

What	Details
The iOS Keychain	See the Apple developer guide for the Keychain Services Concept.
The Android Account Manager	See the Android developer guide for the Account Manager provider.

Subsequent Logins for Inbox-Licensed Users

Use the OAuth access token and refresh token to manage user logins.

After the initial login, users with an Inbox license continue to authenticate to connect their mailbox with the Salesforce Hyperforce server. This authentication obtains and stores an OAuth refresh token and access token, to make requests to Salesforce. This token is tied to the user's Salesforce account.

The OAuth refresh token and access token are stored on the Salesforce Hyperforce servers. The connection makes API calls to obtain an OAuth access token. The access token expires in one hour, by default. A Salesforce admin can configure the expiration time. For example, the admin can set the OAuth refresh token to expire every seven days.

To change the default token expire time, from Salesforce Setup, enter `Connected Apps` in the Quick Find box, and then select **Managed Connected Apps**. Next, select **SalesforceIQ** > **Edit** > **Refresh Token Policy**.

A user's Salesforce password is never stored in the Outlook or Gmail integration or on the Salesforce Hyperforce server. A separate token is obtained for authenticating with the Salesforce Hyperforce servers. This token is a separate Bearer token that is passed in the HTTPS Authorization header.

All network calls to the Salesforce Hyperforce servers are done over TLS 1.2.

The Salesforce admin can block OAuth access at any time. From Salesforce Setup, enter `Connected Apps` in the Quick Find box, and then select **Connected Apps OAuth Usage**.

An individual user can revoke their own Salesforce OAuth token. The Salesforce admin can also revoke OAuth tokens for any user. From your personal settings, view the OAuth tokens in Connections.

Gmail Guidelines

Keep several considerations in mind when connecting with Gmail accounts.

When a user has an Inbox license, the Gmail integration connects their Gmail account using the OAuth 2.0 protocol. The Chrome extension and mobile apps open a page for logging in to your Salesforce account. After this initial login, the Salesforce Hyperforce server obtains and stores an OAuth refresh token and access token for making requests to Google. This token is a single-user token that provides access to that user's Gmail account. The Chrome extension and the Salesforce Hyperforce servers don't use and never store the user's Google password.

Because Inbox mobile apps comprise email and calendar apps, they request full read and write access to the Google email and calendar. They also request read-only access to Google Drive, to support attaching files from Google Drive to emails. Inbox mobile apps request the following Google OAuth permissions.

- https://mail.google.com/
- https://www.google.com/m8/feeds/
- https://www.googleapis.com/auth/userinfo.profile
- https://www.googleapis.com/auth/userinfo.email
- https://www.googleapis.com/auth/calendar

- https://www.googleapis.com/auth/drive.readonly

A user can revoke the Google OAuth token at any time by navigating to the third-party app in the Google account user settings. See Google help for more information about removing third-party apps.

All network calls to Google are made using TLS 1.2 AES-256 cipher. Inbox uses two protocols with Google:

- Gmail API
- Google Calendar API

Exchange Online (Office 365) Guidelines

Review authorization guidelines for Exchange Online (Office 365).

When users have an Inbox license, the Outlook integration connects the Office 365 account using the OAuth 2.0 protocol outlined in this guide. The Outlook add-in and mobile apps open a page for logging in to your Salesforce account. After this initial login, the Salesforce Hyperforce server obtains and stores an OAuth refresh token and access token for making requests to Office 365. This token is a single-user token that provides access to that user's Office 365 account. Salesforce never uses or stores the user's Office 365 password.

Because Inbox mobile apps are email and calendar apps, they request full read and write access to the Office 365 email and calendar. Inbox mobile apps request the following Office 365 permissions.

- Send mail as a user
- Read and write user calendars
- Read and write user contacts
- Read user profiles
- Access mailboxes as the logged-in user via EWS
- Read and write use mail

A user can revoke the Office 365 OAuth token at any time. Go to Office 365, and then select **My Account** and **App Permissions**.

A Microsoft Azure administrator can change the Office 365 OAuth refresh token expiration time by setting it to MaxAgeSingleFactor and MaxAgeMultiFactor in the Microsoft Azure Active Directory.

All network calls between the Outlook add-in or Inbox mobile apps and Office 365 are performed over TLS1.2.

Salesforce uses these protocols to access the email and calendar from Office 365:

What	Protocol	Authentication	Notes
Salesforce Hyperforce servers	EWS	OAuth 2.0 authentication	EWS is a SOAP-based protocol that Microsoft Outlook desktop uses on Windows and Mac. The Salesforce Hyperforce server polls data based on notifications, rather than polling at a regular interval. For more information, see Notification subscriptions, mailbox events, and EWS in Exchange in the Office Dev Center.
Inbox mobile apps	Exchange ActiveSync (EAS) or Office 365 REST API	EAS supports authentication using a username, domain, and password. The Office 365 REST API uses an OAuth 2.0 access token.	Exchange ActiveSync (EAS) is a binary XML-based protocol targeted for mobile apps. The native iOS Mail app and Android Exchange mail app use the EAS protocol. To learn more about Exchange ActiveSync for iOS, view this Apple support topic. Microsoft also has a newer API, called the Microsoft Graph API or Office 365 REST API. The Microsoft Outlook app for iOS and Android can now use the newer Microsoft Graph API for reading and writing email and calendar events.

What	Protocol	Authentication	Notes
			Inbox mobile apps don't use the IMAP protocol.

Microsoft Exchange On-Premises Instances

On-premises Microsoft Exchange instances have important requirements to consider.

Connections with on-premises Exchange instances (2013, 2016, 2019) use legacy authentication, based on username, password, and domain. In this case, the Salesforce Hyperforce server and Inbox mobile apps store the Exchange password. Passwords are encrypted at rest using AES-256 encryption.

Network calls with Exchange are performed using TLS 1.1 or TLS 1.2. The highest TLS protocol supported by the Exchange server determines which TLS protocol is used.

Salesforce uses two protocols to access email and calendar from Exchange:

What	Protocol	Notes
Salesforce Hyperforce servers	EWS	EWS is a SOAP-based protocol used in Microsoft Outlook desktop.
Inbox mobile apps	Exchange ActiveSync (EAS) or Office 365 REST API	Exchange ActiveSync (EAS) is a binary XML-based protocol targeted for mobile apps. The native iOS Mail app and Android Exchange mail app use the EAS protocol.

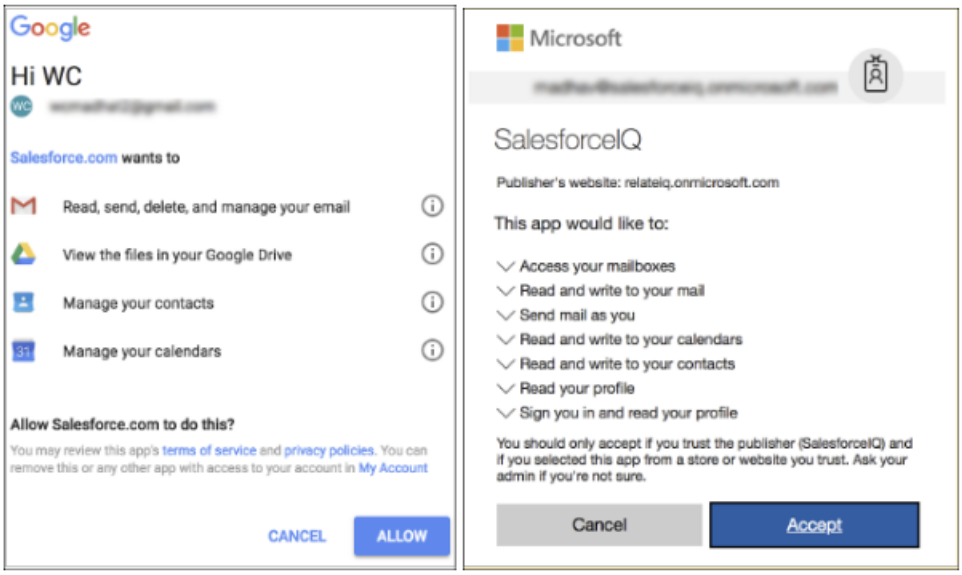
On-premise servers don't support the OAuth 2.0 protocol. Instead, the connection supports NT LAN Manager (NTLMv2) and basic authentication, which require the username, domain, and password for authentication.

Microsoft recently introduced a newer API, called the Microsoft Graph API or Office 365 REST API. Exchange servers that aren't on Office 365 don't support the newer Office 365 REST API or the Microsoft Graph API.

See Microsoft documentation for information about Office 365 APIs.

More About the OAuth Protocol

Salesforce uses the OAuth 2.0 authorization framework to connect with outside services.



When provisioned with an Inbox license, users can connect to the Salesforce platform and to their email provider. Mobile users with a Gmail or Office 365 email account connect using the OAuth 2.0 protocol. After this initial login, the Salesforce Hyperforce server obtains and stores an OAuth refresh token and access token for making requests to the server.

OAuth 2.0 allows third-party applications, such as Salesforce, to obtain access to a service such as Gmail and Office 365 while delegating the login process to that service.

When the user connects to a service, Salesforce opens a new page to the service's login page. This page is the service's own login web page such as:

- https://login.salesforce.com/
- https://accounts.google.com
- https://login.microsoftonline.com

After users log in, they're prompted to grant permissions to the third-party app, in this case, Salesforce. These permissions include:

- Full read and write access to your email.
- Full read and write access to your calendar.

When the user allows permission, the service redirects back to the Salesforce Hyperforce server. The Salesforce Hyperforce server connects to the service and obtains the following:

- An OAuth refresh token
- An OAuth access token

The OAuth access token is used to make authenticated requests to the service, such as fetching email or calendar events. These tokens typically expire after one hour. To obtain another access token, the OAuth refresh token is used. The refresh token typically has a long expiration time or no expiration time.

Salesforce Hyperforce Server Operations

The Salesforce Hyperforce infrastructure stores Inbox data and performs various operations

The Salesforce Hyperforce infrastructure performs these email and calendar operations.

- Fetch and store email headers and HTML body from your mailbox.

- Fetch email headers and HTML body from your Sent Mail.
- Fetch and store calendar events.
- Send email messages with the Send Later feature.
- Create a calendar event with attendees with Insert Availability.

Inbox mobile apps have full read and write access to email and calendar. They perform these operations on email and calendar.

- Reading, composing, and sending email, including attachments
- Marking as read, flagging, moving an email to a folder, archive, trash
- Viewing, creating, and editing calendar events
- Sending RSVPs to calendar events

Mobile Device and Application Management and Inbox

Inbox mobile app has limited support for Mobile Device Management (MDM) and Mobile Application Management (MAM).

Inbox mobile supports MDM Client Certificates for the following:

- A user's Salesforce login.
- When connecting to an Office 365 email account.
- When connecting to a Google email account.

Inbox mobile doesn't support MDM Client Certificates for Exchange 2013 or 2016 email accounts.

For specific vendors:

- Inbox mobile supports Microsoft Intune MAM and MDM for Office 365 email accounts when using Inbox on an iOS device.
- Inbox mobile doesn't support other MDM solutions such as MobileIron, VMWare Airwatch, and others.

Microsoft Intune Support for Inbox Mobile for iOS

Starting with Inbox mobile for iOS version 8.6.0, Salesforce supports adding Inbox mobile for iOS to Microsoft Intune for Mobile Application Management (MAM) and Mobile Data Management (MDM). To add Inbox mobile apps to Intune and to ensure that your Salesforce org is set up correctly, complete the following actions.

Action	Details	Resources
Add the Salesforce Inbox mobile app as a managed app in Intune.	See the Microsoft Documentation for detailed instructions about adding iOS apps to Intune.	Adding an app to Intune requires the App Store link and the bundle ID. **App Store link:** https://apps.apple.com/us/app/salesforce-inbox/id1208232122 **iOS Bundle ID:** com.salesforce.inbox
Create an app protection policy for Inbox mobile app entry in Intune.	See the Microsoft Documentation about creating app protection policies.	
In Azure Active Directory, configure the admin consent setting for Office 365.	To provide a streamlined authorization flow, provide admin consent to Intune. • Navigate to this URL.	Learn more about admin consent

Action	Details	Resources
	• Log in to your Office365 account. A window to grant consent appears. • To grant admin consent permissions to Intune for Salesforce Inbox on behalf of your organization, click **Accept**.	
Turn on Intune support in Salesforce.	Set Salesforce to use Intune for the Inbox mobile app. 1. From Setup, in the Quick Find box, enter *Inbox*, and then select **Setup Assistant**. 2. Turn on Microsoft Intune.	

Mobile App Data Removal

A Salesforce admin can remove a Salesforce or Salesforce Inbox license for a specific user at any time.

After the license is removed, users can no longer log into the app. When a user attempts to log in, the app logs out the user and deletes all data stored locally for that user

An Exchange admin can also initiate an Exchange ActiveSync Remote Wipe.

- In Office 365, go to Office 365 settings to wipe data.
- For on-premises Exchange servers, go to the Exchange Admin Center and wipe data.

After wiping data, if the app is opened, the app connects to Exchange and receives the remote wipe response. It marks the email account as disabled, logs the user out of the app, and deletes all the data stored locally for that user.