

Microsoft 365 Outlook API 完全ガイド

Macから、会社用Outlookのメールを、自分のWebアプリで安全に読めるようになるまでの手順です。

確認日は2026年10月19日です。画面の文言は、表示言語やMicrosoftの更新で少し変わることがあります。押す場所の意味が同じなら、そのボタンを使います。

このファイルだけで完結します

インターネット上の追加ファイルは読み込みません。図はすべてこのページの中に描いてあります。写真が必要な箇所は、公式資料へのリンクを置き、ここでは模式図と書いてあります。

この説明書を作った作業では、会社のMicrosoft 365は変更していません

メールの取得、管理者の同意、秘密情報の作成もしていません。画面を操作するのは、あなたが会社のルールを確認したあとです。

読み方

1. 上から順に読みます。章を飛ばすと、あとで入力する値が決まりません。
2. **青いボタン** は、画面で押す場所です。
3. **黄色の文字** は、そのまま入力する値です。
4. 点線の用語を押すと、日本語の説明が開きます。
5. 図の「拡大」で、模式図を大きく表示します。
6. 各章の「読み終えた」と、最後のチェックリストは、このブラウザに保存されます。

PDFをダウンロード

Cursor実装プロンプト

このページが説明書です。PDFは同じ内容の印刷用、プロンプトは第8章でCursorへ貼る指示です。Microsoftアカウントがなくても読めます。

第1章

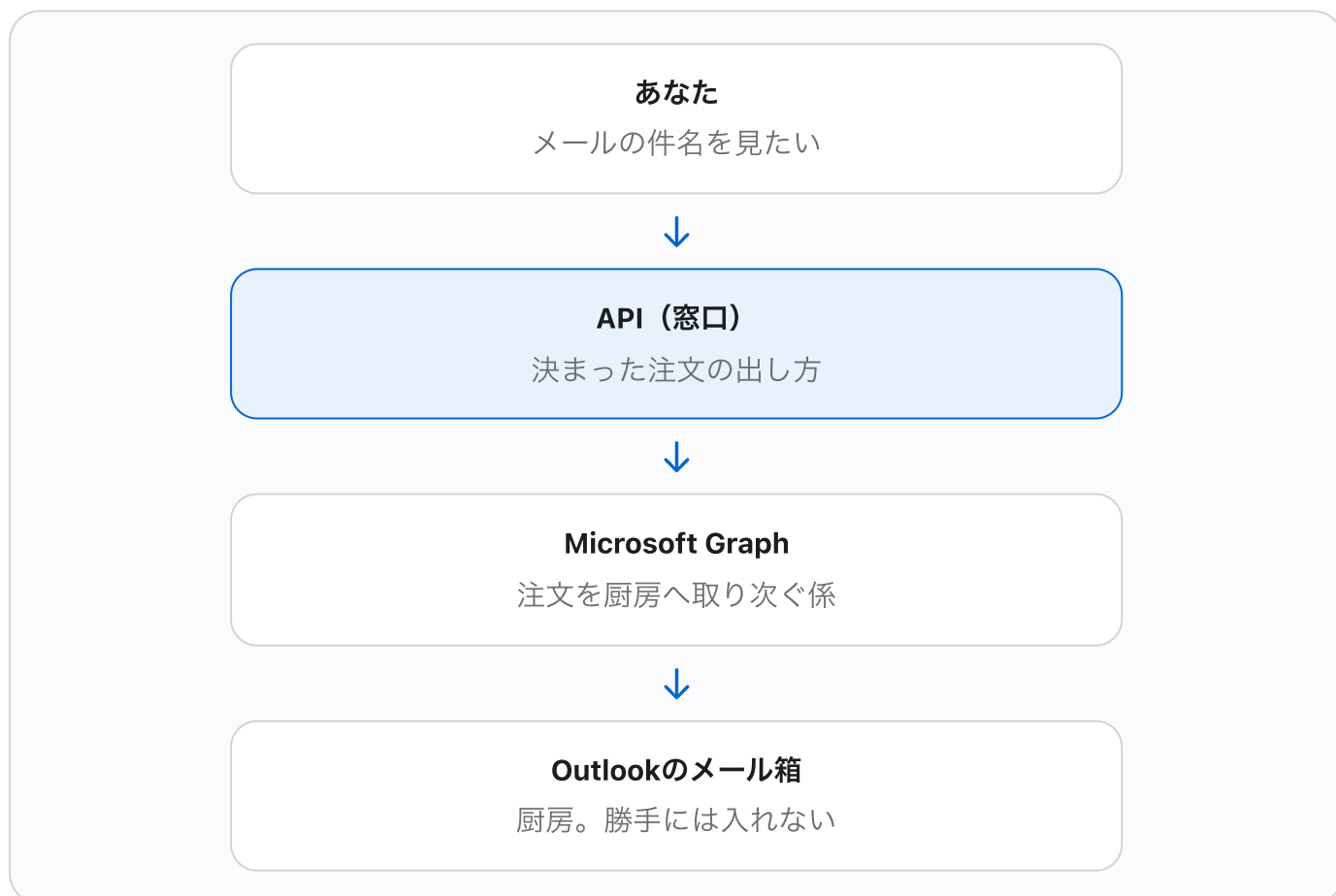
APIとは何か？

☐ この章を読み終えた

[Outlook](#)は、会社のメールを読むソフトです。ブラウザでは Outlook on the web、Macではメールアプリからも開けます。中にあるメールそのものは、あなたのMacの中ではなく、会社のMicrosoft 365にあります。

[Microsoft 365](#)は、Outlook、予定表、ファイルなどをまとめた会社向けのサービスです。メールアドレスが `名前@会社のドメイン` の形なら、それは会社のアカウントです。

[Microsoft Graph](#)は、その会社のサービスへプログラムから届く、Microsoft公式の窓口です。メールを読むときも、この窓口を使います。古いOutlook REST APIという別の窓口は、新しく作るアプリでは使いません。



説明用の模式図 APIは、厨房に勝手に入るかわりに、決まった注文だけを通す窓口です。

APIを使うと何ができるか

この教科書で作るのは、次の1つだけです。サインインした本人の受信トレイから、件名と受信日時を最大10件、自分のWebアプリの画面に表示します。

- できること: 本人のプロフィール確認、受信トレイの件名と受信日時の表示。
- この初期設定ではできないこと: メールの送信、削除、既読への変更、他人のメールを見ること。

APIキーとアクセストークンの違い

街中の説明で「APIキーを発行する」と書かれていても、Outlookのメール読み取りはAPIキー方式ではありません。

名前	やさしい意味	Outlookのメール読み取り
<u>APIキー</u>	アプリに渡す固定の合言葉。漏れると、期限まで使い続けられることが多い。	使いません。
<u>アクセストークン</u>	サインインした人の代わりに、短い時間だけ通行できる通行証。	これを使います。画面やログには出しません。

この章のゴール

メールはMicrosoftの中にあり、Graphという窓口へ、短い通行証を見せて読みにいく、と説明できれば完了です。

第2章

接続の全体像

☐ この章を読み終えた

パスワードをWebアプリに保存してOutlookへ入る方法は使いません。あなたはMicrosoftのログイン画面でサインインし、Webアプリはあとから通行証だけを受け取ります。

1. 利用者

会社のアカウントを持っている人

↓ ボタンを押す

2. 自分のWebアプリ

Macの中で練習用に動かす

↓ ログイン画面へ移動

3. Microsoftのログイン認証

login.microsoftonline.com

↓ 何を読んでよいか確認

4. アクセス許可

本人の基本メール情報だけ

↓ 通行証を持って問い合わせ

5. Microsoft Graph API

graph.microsoft.com

↓ 件名と受信日時だけ返す

6. Outlookのメール取得

受信トレイ。送信や変更はしない

↓ 画面用の文字だけ返す

7. 自分のアプリに表示

件名と受信日時

説明用の模式図 上から下へ進みます。パスワードが自分のアプリに保存される段はありません。

認証と、メール取得の通信方向

認証の往復

1. ブラウザ → 自分のアプリ: サインインを押す
2. 自分のアプリ → ブラウザ → Microsoft: 認可のお願い
3. あなた → Microsoftの画面: 会社アカウントでサインイン
4. Microsoft → 自分のアプリ: 一度きりの認可コード
5. 自分のアプリのサーバー → Microsoft: コードを通行証へ交換

メール取得の往復

1. 自分のアプリのサーバー → Graph: 通行証を添えて受信トレイを要求
2. Graph → Outlook: 本人の受信トレイを参照
3. Graph → 自分のアプリ: 件名と受信日時
4. 自分のアプリ → ブラウザ: その2項目だけを表示

説明用の模式図 左がログインの通信、右がメールを読む通信です。通行証は右の通信で使います。

順 番	送り手	受け手	渡るもの	渡さないもの
1	ブラウザ	自分のWebアプリ	サインインボタンの操作	パスワード
2	Webアプリ	Microsoft	アプリID、戻り先、読みたい範囲、合言葉、PKCEの問い	クライアントシークレット、PKCEの答え
3	あなた	Microsoftの画面	会社アカウントと、会社が求める追加確認	自作アプリの入力欄へのパスワード
4	Microsoft	Webアプリ	認可コードと、同じ合言葉	URLに載ったアクセストークン
5	Webアプリのサーバー	Microsoft	認可コード、PKCEの答え、クライアントシークレット	ブラウザから見えるシークレット
6	Microsoft	サーバー	アクセストークンとリフレッシュトークン	画面表示やログ
7	サーバー	Microsoft Graph	通行証と、受信トレイの取得要求	送信や削除の要求
8	Graph	サーバー	件名と受信日時	この初期設定では本文
9	サーバー	ブラウザ	件名と受信日時の表示	通行証

この章のゴール

「パスワードはMicrosoftへ。通行証はサーバーの中。画面に出るのは件名と日時」と言えば完了です。

事前準備

☐ この章を読み終えた

必要なもの	判断のしかた
Mac	SafariかChromeができれば足ります。この説明書の確認はMacを前提にしています。
会社の Microsoft 365	普段、会社のメールアドレスでOutlookを開いていること。
Microsoft Entra ID	会社のMicrosoft 365に含まれます。別に契約を買う手順はこの教科書にありません。
アプリを登録 できる権限	公式手順を完了できる最小の管理者ロールは、クラウド アプリケーション管理者です。一般の社員も登録できる会社と、できない会社があります。
メールを読む 同意	確認日のMicrosoft推奨設定では、メールを読む許可は利用者本人の同意だけでは足りず、管理者の同意が必要です。
Node.js	第8章でWebアプリを作るときに使います。ターミナルで <code>node -v</code> と入力し、バージョンが表示されることを確認します。表示されない場合は、nodejs.org のLTSを入れてから先へ進みます。
Cursor	コードを自分で暗記する必要はありません。第8章のプロンプトを貼ります。

一般の社員がアプリを登録できないとき

会社は「Users can register applications（ユーザーはアプリケーションを登録できる）」を無効にできません。場所は Entra ID、ユーザー、ユーザー設定です。無効なら、新規登録が失敗します。その場合は作業を止め、クラウド アプリケーション管理者かアプリケーション開発者ロールを持つ人に登録を依頼します。

Microsoft 365の契約と、Azureの課金

Microsoft 365

Outlookのメール箱を使う契約です。会社がすでに払っています。この教科書のアプリ登録と、本人のメール読み取りは、この契約の中の機能です。

Azureの課金

仮想マシンやApp Serviceなど、Azureの資源を新しく作ると料金が発生します。この教科書では、その資源を作りません。

確認日の公式一覧で、呼び出し回数に応じてAzureへ課金されるGraph APIは、SharePointとOneDriveのassignSensitivityLabelです。メールの `/me` と `/me/mailFolders/inbox/messages` はこの一覧にありません。TeamsのAPIは2025年8月25日から従量課金の対象外になっています。

断定できないこと

会社独自の社内ルールや、政府向け・中国向けなど特別なクラウドを使っている場合の住所は、この説明書では決められません。普段のOutlookが `outlook.office.com` で開けるなら、この教科書のグローバル向けの住所を使います。開けない場合は管理者に、どのクラウドかを聞いてから止めます。

Microsoft Entraでアプリ登録

☐ この章を読み終えた

開くページは <https://entra.microsoft.com/> です。アプリの登録は、プログラムをMicrosoftへアップロードする作業ではありません。会社の名簿に「このWebアプリが、あとでサインインを依頼してよい」と記録する作業です。

1

会社のアカウントでログインする

説明用の模式図 実際のログイン画面の写真ではありません。

開くページ <https://entra.microsoft.com/>

入力する値 会社のメールアドレス。個人の @outlook.com、@hotmail.com、@gmail.com は使いません。

押す場所 **次へ**、そのあとMicrosoftが示すパスワードと、会社の追加確認。

成功すると 画面上部に「Microsoft Entra 管理センター」または Microsoft Entra admin center と出ます。

目的の会社のディレクトリを選ぶ

押す場所

複数の組織に所属しているときだけ、上部の歯車 **設定**、それから **ディレクトリ + サブスクリプション**。

選ぶもの

会社のメールが属するディレクトリ。個人用のディレクトリのまま登録しません。

成功すると

画面の組織名が、会社名になっています。所属が1つなら、この操作は出ません。そのまま次へ進みます。

アプリの登録を開く



説明用の模式図 左のメニューでアプリの登録を開き、一覧の「新規登録」へ進みます。実際のボタン位置は更新で前後します。

押す場所

左の **ID**（英語では Identity）、**アプリケーション**（Applications）、**アプリの登録**（App registrations）。

成功すると

一覧と、**新規登録**（New registration）が見えます。

止めるとき

メニューが無い、または権限エラーになる。クラウド アプリケーション管理者へ依頼し、自分で権限を変更しません。

4

新規登録を選ぶ

押す場所

新規登録

成功すると

「アプリケーションの登録」というフォームが開きます。

5

アプリ名を入力する

入力する値

Outlookメール表示（練習）

理由

会社の管理画面に残る名前です。練習だと自分で分かる名前にします。顧客へ公開する製品名にはしません。

成功すると

名前の欄に、今入力した文字が表示されています。

サポート対象アカウントを選ぶ

● ● ● アプリケーションの登録

名前: Outlookメール表示（練習）

選択中: この組織ディレクトリのみに含まれるアカウント

任意の組織ディレクトリ内のアカウント

個人用 Microsoft アカウントを含む広い選択肢

リダイレクト URI: 空欄のまま

登録

説明用の模式図 青枠が、会社用として選ぶ項目です。リダイレクトURIは、この画面では空欄です。

選ぶもの

「この組織ディレクトリのみに含まれるアカウント」。英語では Accounts in this organizational directory only。

選ばないもの

任意の組織ディレクトリ、個人用Microsoftアカウント。会社のOutlookだけを対象にするためです。

リダイレクト URI

この画面では空欄のままにします。公式の登録手順です。戻り先は第5章で、種類をWebにしてから登録します。

登録を完了する

押す場所

登録 (Register)

成功すると

「概要」が開きます。表示名が Outlookメール表示（練習）です。アプリケーション (クライアント) ID、オブジェクト ID、ディレクトリ (テナント) ID が見えます。

次に使うID

アプリケーション (クライアント) ID と、ディレクトリ (テナント) ID。オブジェクト ID はコピーしません。

公式ページの読み分け

一部の公式ページは、アプリケーションIDをオブジェクトIDと呼ぶ書き方をしています。登録手順のページは、2つを別の値として説明しています。画面のラベルをそのまま見て、ラベルどおりにコピーします。

必要なIDを取得する

☐ この章を読み終えた

● ● ● 概要

アプリケーション (クライアント) ID	コピーする
オブジェクト ID	使わない
ディレクトリ (テナント) ID	コピーする

説明用の模式図 概要ページでコピーするのは、上と下の2つです。値の実物はこの図に入れていません。

テナントIDが概要に見当たらないときは、Entra ID のトップの概要にも同じ「ディレクトリ (テナント) ID」があります。ラベルの文字が一致する方をコピーします。

項目	意味	取得場所	保存方法	公開してよいか
アプリケーション (クライアント) ID	このアプリの識別番号。 <u>クライアント ID</u> と同じものです。	アプリの概要	あとで作る <code>.env.dev</code> の <code>CLIENT_ID</code>	パスワードではありません。それでもチャットや公開ページには貼りません。
ディレクトリ (テナント) ID	会社のディレクトリの識別番号。	アプリの概要、または Entra IDの概要	<code>.env.dev</code> の <code>TENANT_ID</code>	クライアントIDと同じ扱いです。
リダイレクトURI	ログイン後に戻ってくる、自分のアプリの住所。	次の手順で自分が進めて登録します	Entraと <code>.env.dev</code> の両方。文字は完全一致	練習用の localhost は、インターネットへ公開した住所ではありません。
クライアント シークレット の値	サーバー用アプリのパスワード。	証明書とシークレット。作成直後の「値」だけ	<code>.env.dev</code> の <code>CLIENT_SECRET</code> 。Git へ入れない	公開しません。この説明書、チャット、スクリーンショットに写しません。
シークレット ID	シークレットを管理するための番号。	作成後の一覧に残る	保存しない	値の代わりには使えません。

リダイレクトURIをWebとして登録する

● ● ● 認証

概要

認証

証明書とシークレット

APIのアクセス許可

プラットフォームを追加

Web

http://localhost:3000/auth/redirect

http://localhost:3000/

暗黙的な許可のチェックは付けません。

説明用の模式図 選ぶタイルは Web です。シングルページ アプリケーションは選びません。

1

押す場所

左の管理の中の **認証** (Authentication)。

成功すると

プラットフォームの構成が開きます。

2

押す場所

プラットフォームを追加 (Add a platform)。

選ぶタイル

Web。サーバーで動くWebアプリ用です。

3

入力する値

1件目 `http://localhost:3000/auth/redirect`

2件目 `http://localhost:3000/`

入力の規則

httpsにはしません。大文字にしません。ポートは3000です。1件目の末尾にスラッシュは付けません。2件目は末尾スラッシュが必要です。パスの無い住所は、Microsoftが応答時にスラッシュを足すためです。

チェックしない

暗黙的な許可とハイブリッド フローの、アクセストークンとIDトークン。

押す場所

構成 (Configure)。

成功すると

Webの欄に、上の2件がそのまま表示されます。

クライアント シークレットが必要になる条件

必要

この教科書のWebサーバー型です。秘密はMacの中のサーバーだけが持ちます。練習ではクライアント シークレットを使います。公式は、本番では証明書の方が安全だと案内しています。

不要

ブラウザだけで動くSPA、スマホアプリ。これらは公開クライアントです。シークレットを置く場所が安全でないため、PKCEだけで進めます。この教科書の初期構成では選びません。

1

押す場所

証明書とシークレット、クライアント シークレット、
新しいクライアント シークレット。

入力する値

説明は `local-dev`。有効期限は6か月。公式の上限は24か月で、推奨は12か月未満です。

押す場所

追加

成功すると

「値」列に文字列が1回だけ出ます。その場で `.env.dev` に保存します。シークレット ID は保存しません。

値は二度と表示されません

ページを離れたあとに見えないのは故障ではありません。見損なったら、新しいシークレットを作り、古いものは削除します。値をこの説明書へ書き写さないでください。図にも載せていません。

保存先のファイルは、まだ中身がプレースホルダの見本です。本物の値への差し替えは、第8章でアプリケーションフォルダを作ったあと、あなた自身が行います。

```
CLOUD_INSTANCE=https://login.microsoftonline.com/  
TENANT_ID=00000000-0000-0000-0000-000000000000  
CLIENT_ID=00000000-0000-0000-0000-000000000000  
CLIENT_SECRET=replace-with-the-secret-value  
REDIRECT_URI=http://localhost:3000/auth/redirect  
POST_LOGOUT_REDIRECT_URI=http://localhost:3000/  
GRAPH_API_ENDPOINT=https://graph.microsoft.com/  
EXPRESS_SESSION_SECRET=replace-with-a-long-random-string
```

コピー

CLOUD_INSTANCE と GRAPH_API_ENDPOINT の末尾の / は残します。消すと、テナントIDの直前で住所がつながります。EXPRESS_SESSION_SECRET は、シークレットとは別の、自分で作った長いランダム文字列です。

Microsoft Graphのアクセス許可

☐ この章を読み終えた

初期構成は、委任されたアクセス許可です。アプリが自分の権限で全員のメールを読む方式ではありません。サインインした本人の範囲だけです。

最小権限に合わせ、件名と受信日時だけを読む許可を選びます。確認日の「メッセージ一覧」の公式ページでは、仕事用アカウントで最も小さい委任の許可は Mail.ReadBasic です。Mail.Read は本文も読める、ひとつ強い許可です。

許可	種類	公式の意味	初期設定
User.Read	委任	サインインし、本人のプロフィールを読む	残す。登録時から付いていることが多い
Mail.ReadBasic	委任	本人のメールを読む。本文、プレビュー、添付、拡張プロパティは除く	追加する。この教科書の目標にちようど合う
Mail.Read	委任	本人のメールボックスを読む。本文も対象	最初は追加しない。本文が必要になったとき、Mail.ReadBasic と入れ替える
Mail.ReadWrite	委任	作成、読み取り、更新、削除	追加しない
Mail.Send	委任	メールを送信する	追加しない
Mail.ReadBasic.All など	アプリケーション	サインイン無しで組織のメールを読む	追加しない

権限一覧の「管理者の同意が必要」列は、User.Read、Mail.ReadBasic、Mail.Read の委任では No です。ただし会社の同意ポリシーが優先されます。確認日のMicrosoft管理ポリシーでは、利用者は Mail.ReadBasic と Mail.Read に自分では同意できません。管理者の承認画面が出るのは、設定ミスとは限りません。

● ● ● APIのアクセス許可

Microsoft Graph / 委任されたアクセス許可

検索: Mail.ReadBasic

チェックする: Mail.ReadBasic Read user basic mail

チェックしない: Mail.Read / Mail.Send / Mail.ReadWrite

アクセス許可の追加

説明用の模式図 検索語は英語の Mail.ReadBasic です。似た名前が並ぶので、最後の Basic まで一致するものを1つだけ選びます。

1

押す場所

APIのアクセス許可

確認する

User.Read があり、種類が委任または Delegated であること。無ければ、次と同じ手順で User.Read を追加します。

2

押す場所

アクセス許可の追加、 Microsoft Graph、 委任されたアクセス許可

。

選ばないタブ

アプリケーションの許可。

3

入力する値

検索ボックスに Mail.ReadBasic

チェックする

Mail.ReadBasic が1つ。表示テキストは Read user basic mail。

押す場所

アクセス許可の追加

成功すると

一覧が User.Read と Mail.ReadBasic です。両方とも委任です。状態は、まだ同意前なら未付与です。

管理者の同意

あなたがクラウド アプリケーション管理者でない場合、同意ボタンは押しません。管理者へ、次の文面を渡します。クライアント シークレットは書きません。

件名: 練習用アプリの委任アクセス許可の同意をお願いします

コピー

アプリ名: Outlookメール表示 (練習)

アプリケーション (クライアント) ID: (概要のクライアントIDだけ)

必要な許可: 委任の User.Read と Mail.ReadBasic

使わない許可: Mail.Send、Mail.ReadWrite、アプリケーションの許可

目的: サインインした本人の受信トレイの件名と受信日時を、本人の練習用Webアプリに表示します。送信、削除、変更はしません。

管理者の公式手順は、Entra ID、アプリの登録、対象アプリ、APIのアクセス許可を開き、内容を確認してから Grant admin consent を押す、です。日本語画面では「管理者の同意を与えます」の系統のボタンです。同意できる最小ロールは、この委任の許可ならクラウド アプリケーション管理者です。成功すると、状態列が組織に対して付与済みになります。

全社員へ開くかどうかは管理者の判断です

組織全体への同意は、既定では利用者全員がそのアプリへサインインできる状態になります。練習アプリを本人だけにしたい場合は、管理者がユーザー割り当てを必須にし、本人だけを割り当てます。

OAuth認証

☐ この章を読み終えた

使う方式は、[OAuth 2.0](#)の認可コードフローに、PKCEを足したものです。認可コードは「通行証と交換できる、すぐ期限が切れる引換券」です。PKCEは、その引換券を盗まれても、発行時の答えを知るアプリ以外が交換できないようにする仕組みです。

- 1 アプリがランダムな「答え」(code_verifier)を作り、セッションに隠します。
- 2 答えをSHA-256で変換した「問い」(code_challenge)だけを、Microsoftへ送ります。方式は S256 です。
- 3 同時に state という合言葉も送ります。戻ってきた合言葉が同じかを確認し、別のサイトからの戻りを避けます。
- 4 あなたがMicrosoftでサインインし、許可に同意します。
- 5 Microsoftは引換券（認可コード）を、登録済みの戻り先へPOSTします。この推奨構成では、住所欄にコードを付けません。
- 6 サーバーが、引換券と答えとクライアント シークレットを、トークン窓口へ送ります。
- 7 Microsoftが通行証を返します。サーバーのセッションだけが保持します。

説明用の模式図 問いだけが先に出て、答えは交換のときまでサーバーに残ります。

認可の窓口は `https://login.microsoftonline.com/{テナントID}/oauth2/v2.0/authorize` です。通行証の窓口は同じホストの `/oauth2/v2.0/token` です。会社専用アプリでは、common ではなくテナントIDを使います。

Webサーバー型とSPA型

Webサーバー型 これを使う

Entraのタイルは Web。ライブラリは @azure/msal-node。シークレットはサーバーだけ。戻り先は `http://localhost:3000/auth/redirect`。通行証はブラウザへ出しません。

SPA型 今回は使わない

Entraのタイルはシングルページ アプリケーション。ライブラリは @azure/msal-browser。シークレットは使いません。ブラウザが直接トークン窓口へ話します。

説明用の模式図 タイルを取り違えると、ログイン画面のあとでエラーになります。コードの種類とタイルをペアにします。

守り方	この練習での具体
state	ログイン開始時に作り、戻りで同じ値かを見ます。違うときは通行証へ交換しません。
トークン管理	サーバーのセッションに置きます。ブラウザの <code>localStorage</code> には置きません。画面のHTMLにも書きません。
HTTPS	<code>localhost</code> の <code>http</code> は開発中だけの例外です。本番の戻り先は <code>https</code> にします。
秘密の保存	<code>.env.dev</code> に置き、 <code>.gitignore</code> でGitの対象外にします。ソースコードへ直接書きません。
期限	通行証の <code>expires_in</code> は秒数です。公式の例では約60分前後の値があります。固定の1時間とは限りません。延長はリフレッシュトークンで、MSALの <code>acquireTokenSilent</code> に任せます。

実装は手書きのHTTPにしない

上の流れを理解したうえで、交換処理はMSALに任せます。公式も、本番では認証ライブラリを使うよう案内しています。

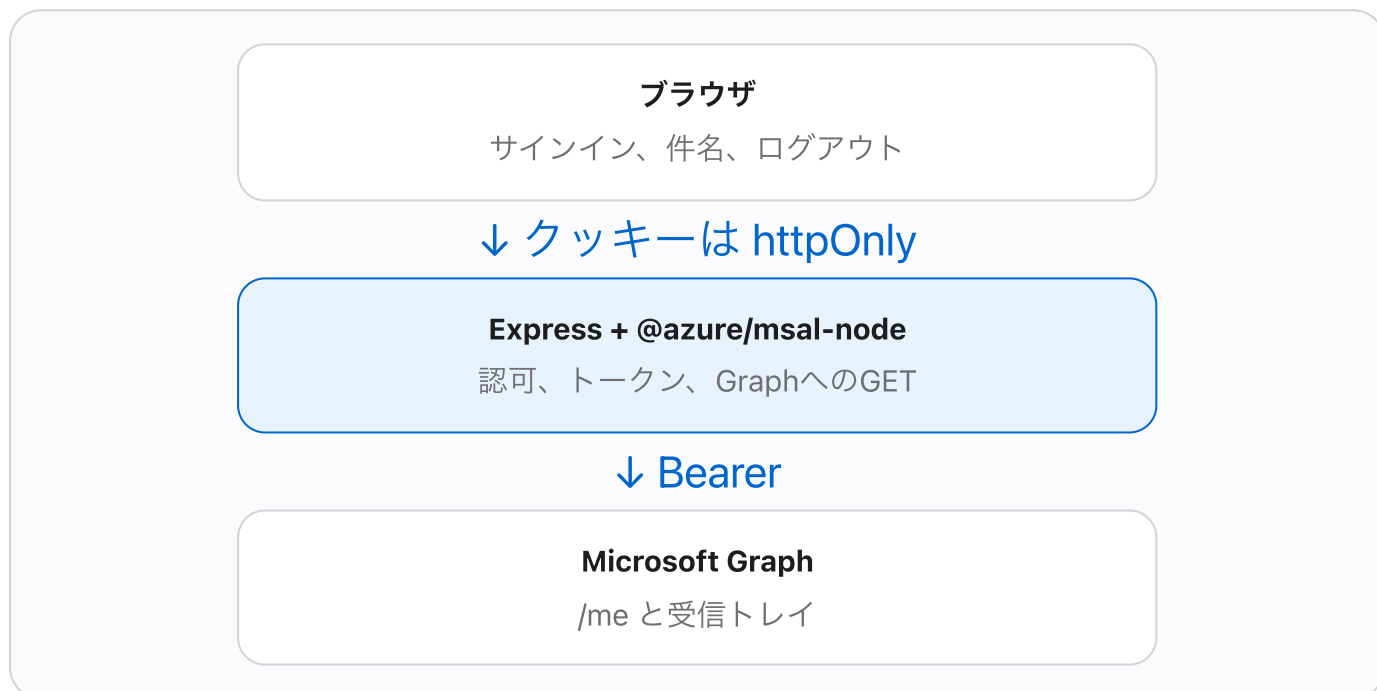
CursorでWebアプリに接続する

☐ この章を読み終えた

接続先の置き場所

既存のアプリが Swift や iOS だけの場合、そこに Outlook 接続を足しません。練習用の Web アプリは、別の新しいフォルダに作ります。

フォルダ名の例は `outlook-mail-reader` です。すでに動いている別のアプリのフォルダや、会社の共有フォルダの中には作りません。



説明用の模式図 通行証を扱うのは、真ん中のサーバーだけです。

Cursorへ貼る

1. Cursorで、新しいウィンドウを開きます。フォルダはまだ空で構いません。
2. Cursor実装プロンプトを開き、中の指示をコピーします。
3. 「ここから下を、新しいCursorチャットへ貼る」の枠内を、チャットへ貼ります。
4. Cursorが新しいフォルダにファイルを作るのを待ちます。既存の別アプリにファイルが増えていたら、その変更は採用しません。
5. できた `.env.dev.example` を複製し、名前を `.env.dev` にします。
6. プレースホルダを、第5章で保存した値に差し替えます。この差し替えはチャットでは行いません。

プロンプトがCursorに頼む機能は、次の7つです。

- Microsoftでサインインするボタン
- POSTの認証コールバック
- サーバーセッションでのトークン取得
- Graphの /me への接続
- 受信トレイの件名と受信日時の読み取り
- トークンを隠したエラー表示
- セッション破棄を含むログアウト

接続の要所

スコープはこの2つです。MSALが openid、profile、offline_access を足すのは、公式チュートリアルにも書かれている既定の動作です。

```
const scopes = ["User.Read", "Mail.ReadBasic"];
```

コピー

メール取得は GET だけです。受信日時は UTC で届くので、画面では東京の時刻に直します。

```
const url = process.env.GRAPH_API_ENDPOINT + "v1.0/me/mailFolders/inbox/messages";
const response = await axios.get(url, {
  headers: { Authorization: `Bearer ${accessToken}` },
  params: {
    "$select": "subject,receivedDateTime",
    "$top": 10,
    "$orderby": "receivedDateTime desc"
  }
});
```

コピー

ソースコードへ秘密を書かない

CLIENT_SECRET の右辺に、Entraで表示された値を貼ったファイルは、その時点で不合格です。値は .env.dev だけに置きます。

第9章

接続テスト

☐ この章を読み終えた

テストは、アプリのフォルダだけで行います。ターミナルでそのフォルダを開き、次を実行します。

```
npm install
npm start
```

コピー

ブラウザで `http://localhost:3000` を開きます。成功条件は上から順に見ます。1つ失敗したら、次へ進まず第10章へ移ります。

localhost:3000/mail

サインイン中: 自分の表示名

件名の例	2026/10/19 14:00
------	------------------

(件名なし)	2026/10/19 13:12
--------	------------------

ログアウト

本文、送信ボタン、削除ボタンは無い。

説明用の模式図 成功画面の見本です。実在のメールは表示していません。

1. Microsoftログイン成功: 会社のアカウントで戻り、アプリのトップがエラーでは無い。
2. 必要な権限の同意: 同意画面に出るメール関連が、基本の読み取りである。送信や、すべてのメールボックスの読み取りが含まれていたなら、その場でキャンセルし、第6章の許可を見直します。
3. /me 取得成功: 自分の表示名、または会社のメールアドレスがアプリに出る。
4. /me/mailFolders/inbox/messages 取得成功: 受信トレイの一覧がエラーでは無い。
5. 画面に件名と受信日時が出る。日時は日本時間で読める。
6. 不要な送信、変更、削除が0件である。画面に送信ボタンが無く、Outlookの送信済みにもテスト由来の新着が無い。

ログアウトもテストです

ログアウト後にメール一覧を開き直すと、もう一度サインインを求められます。求められない場合は、セッションが残っています。

エラー対処

☐ この章を読み終えた

エラー番号を、この順で見ます。番号が分かるときは、その段だけを直します。

画面の文言をそのまま読む

AADSTSで始まる番号、または401 / 403



管理者の承認

許可を足さず、第6章の依頼文を送る

↓ 番号が別なら

戻り先の不一致

EntraのWeb欄と .env.dev を1文字ずつ比べる

↓ 番号が別なら

401 または期限

一度ログアウトし、サインインし直す

↓ 番号が別なら

403

通行証の範囲に Mail.ReadBasic が含まれるか確認する

説明用の模式図

上から順に切り分けます。トークン自体は人に送られません。

見えるもの	意味	直す場所
管理者の承認が必要です。AADSTS90094、AADSTS65001	本人または管理者が、まだ同意していません。会社の推奨ポリシーではメール読み取りは管理者同意です。	許可を増やしません。管理者へ第6章の文面を送ります。
AADSTS50011	戻り先が、登録した住所と一致しません。	http、localhost、ポート3000、パス、末尾スラッシュ、大文字小文字を比べます。
AADSTS9002326	ブラウザからの通行証交換なのに、戻り先がSPAとして登録されていません。	今回のサーバー型では、タイルをWebのままにします。SPA用のコードへ変更したときだけ、タイルもSPAにします。
AADSTS700025	公開クライアントなのにシークレットを送っています。	Webサーバー型ではWebタイルを使います。SPAではシークレットを送りません。
401 Unauthorized	通行証が無い、期限切れ、または宛先が違います。	Authorizationヘッダーが Bearer かを見ます。値はログに出しません。期限切れならサインインし直します。
403 Forbidden	通行証はあるが、操作に対して許可が足りません。	委任の Mail.ReadBasic と管理者の同意を確認します。アプリケーション許可は足しません。
AADSTS7000215	クライアント シークレットの値が違います。	シークレット ID ではなく「値」を入れ直します。見損なった値は再表示できないので、新しいシークレットを作ります。
AADSTS7000222	シークレットの期限が切れています。	新しいシークレットを作り、.env.devだけを更新します。

見えるもの	意味	直す場所
AADSTS700016	そのテナントにアプリが見つかりません。	テナントIDとクライアントIDの取り換え、別の会社ディレクトリへのログインを確認します。
AADSTS50020	そのアカウントは、サインイン先のテナントにありません。	個人のMicrosoftアカウントではなく、会社アカウントで入り直します。第4章のアカウント種類が「この組織のみ」であることも確認します。
AADSTS50105	アプリがユーザー割り当て制なのに、自分が割り当てられていません。	管理者に、自分のアカウントの割り当てを依頼します。
InefficientFilter	並び替えとフィルターの組み合わせが複雑すぎます。	\$filter は付けません。\$orderby だけで失敗するときは、並び替えを外して取得し、画面側で日時順にします。

Microsoft 365アカウントの違い

会社のアカウントは、組織のディレクトリに属します。@outlook.com や @hotmail.com の個人アカウントは別です。第4章で「この組織のみ」を選んだアプリには、個人アカウントでは入れません。会社のメールを個人アカウントへ転送して読む方法も、この教科書では使いません。

安全な運用

☐ この章を読み終えた

- 最小権限: 一覧に残す委任は User.Read と Mail.ReadBasic です。本文が必要になるまで Mail.Read には上げません。
- 秘密情報: シークレットの値、通行証、認可コード、セッション秘密は .env.dev とサーバーセッションの中だけです。
- ログ: MSALの piiLoggingEnabled は false です。エラー画面には番号と状態だけを出します。
- 練習の終了: 使い終わったらログアウトし、ターミナルでサーバーを止めます。

アクセス権の取り消し

自分が同意した分

myapps.microsoft.com のタイルから Manage your application

↓ 組織全体の同意はここから外せないことがある

管理者

エンタープライズ アプリケーション、アクセス許可、管理者の
同意、取り消し

↓ アプリ自体をやめるとき

アプリの登録の概要で削除

削除の結果を読んでから確定する

説明用の模式図 利用者と管理者で、取り消せる範囲が違います。

myapps で利用者が取り消せるのは、利用者自身が同意した許可です。組織全体の管理者同意は、クラウド アプリケーション管理者以上が、エンタープライズ アプリケーションの「管理者の同意」タブから取り消します。ユーザー同意タブは、確認日の公式説明ではポータルから取り消せません。

アプリ登録の削除は、概要の Delete です。公式は、完全に消す前に無効化も検討するよう案内しています。エンタープライズ アプリケーションの削除は30日間復元できます。アプリ登録の削除画面では、表

示された結果を読んでから確定します。通行証は削除の瞬間に必ず無効になるとは限らないので、サーバーも止めてセッションを捨てます。

会社の管理者への確認事項

- 一般社員がアプリを登録してよいか。
- Mail.ReadBasic の委任に、管理者同意を出してよいか。
- 練習アプリを本人だけに制限するか。
- 使っているのが通常のグローバルクラウドか。
- 練習が終わったアプリを、いつ削除するか。

本番公開前チェックリスト

localhost の練習が成功しただけでは、インターネットへ公開しません。公開前に、次がすべて揃ってから管理者と再確認します。

- 戻り先が https である。
- クライアント シークレットではなく、公式が本番で推奨する証明書へ切り替えている。
- セッションのクッキーが secure である。
- シークレットと .env.dev がGitの履歴に一度も入っていない。
- ログ収集サービスへ通行証が送られない。
- 許可が User.Read と Mail.ReadBasic の委任のみである。
- 送信、削除、既読変更のコードが存在しない。

設定完了チェックリスト

最後に、上から順に確認する

- ☐ 会社のアカウントで Entra 管理センターを開けた
- ☐ アプリ名 Outlookメール表示（練習）で登録した
- ☐ アカウントの種類は、この組織ディレクトリのみ
- ☐ クライアントIDとテナントIDを .env.dev に保存した

- ☐ オブジェクトIDは使っていない
- ☐ Webプラットフォームに、指定の戻り先が2件ある
- ☐ シークレットの値は .env.dev だけにあり、チャットへ貼っていない
- ☐ 委任は User.Read と Mail.ReadBasic だけ
- ☐ Mail.Send、Mail.ReadWrite、アプリケーション許可は無い
- ☐ 管理者の同意が必要なら、依頼して付与済みになった
- ☐ Microsoftログインが成功した
- ☐ /me で本人の表示が成功した
- ☐ 件名と受信日時が画面に出た
- ☐ 送信、変更、削除は0件
- ☐ ログアウト後は、もう一度サインインが必要

用語集

本文の点線を押すと、ここと同じ説明がダイアログで開きます。印刷したPDFでも読めるように、ここに全文を置いています。

Outlook

Microsoft 365のメール機能です。この教科書では、会社のメールアドレスで使うOutlookを指します。

Microsoft 365

会社向けのMicrosoftのサービス一式です。Outlookのメール箱は、この契約の中にあります。

Microsoft Graph

メールや予定表などへ、プログラムから公式にアクセスする窓口です。住所は <https://graph.microsoft.com> です。

API

ソフト同士が決まった形で依頼と応答を交わす窓口です。厨房に直接入るのではなく、注文票を出すことに似ています。

APIキー

固定の合言葉でAPIを使う方式です。この教科書のOutlookメール読み取りでは使いません。

アクセストークン

サインインした人の代わりに、短い時間だけAPIを呼ぶ通行証です。Bearerという種類で、Graphへ提示します。

Microsoft Entra ID

会社のアカウントとアプリの登録を管理する、MicrosoftのIDサービスです。以前はAzure Active Directoryと呼ばれていました。

テナント

会社ごとのディレクトリです。ディレクトリ (テナント) ID は、そのディレクトリの識別番号です。

クライアントID

登録したアプリの識別番号です。画面ではアプリケーション (クライアント) ID と表示されます。オブジェクトIDとは別です。

リダイレクトURI

ログイン後に利用者が戻ってくる、自分のアプリの住所です。登録した文字列と、アプリが送る文字列が完全一致している必要があります。

クライアント シークレット

サーバー型アプリがMicrosoftへ自分を証明するパスワードです。作成直後の「値」だけが秘密で、一覧に残るシークレットIDは秘密の値ではありません。

委任されたアクセス許可

サインインした本人の権限の範囲で、アプリが代わりに操作する許可です。英語では Delegated permissions です。

アプリケーションのアクセス許可

サインインした人がいなくても、アプリ自身の権限で組織のデータへ届く許可です。この教科書の初期設定では使いません。

最小権限の原則

目的に必要な許可だけを付ける考え方です。件名と日時だけなら Mail.ReadBasic、本文も必要なら、その時点で Mail.Read に上げます。

OAuth

パスワードをアプリに渡さず、利用者がIDサービスでサインインし、アプリには期限付きの通行証を渡す標準の手順です。

PKCE

認可コードの交換を、コードを発行したアプリだけに限定する仕組みです。読み方はピクシーに近いです。問いの方式は S256 を使います。

state

ログイン開始時にアプリが作り、戻ってきたとき同じかを確認する合言葉です。別のサイトから偽の戻りを受け取るのを防ぎます。

SPA

ブラウザの中だけで動く Webアプリです。Entraではシングルページ アプリケーションのタイルを使い、クライアント シークレットは使いません。

MSAL

Microsoft Authentication Library の略です。認可コード、PKCE、トークンの更新を、自作のHTTPではなく公式のライブラリに任せるためのものです。

確認日 2026年10月19日

公式資料

- Microsoft Graph の概要
- 認証と承認の基本
- アプリケーションを登録する
- ユーザーの代わりにアクセスする
- 認可コードフローとPKCE
- リダイレクトURIの制限
- Node.js Webアプリのチュートリアル
- アクセス許可のリファレンス
- メッセージ一覧
- アプリの同意ポリシー
- 管理者の同意
- エラーコード
- 従量課金APIの一覧
- アプリ登録の削除
- マイアプリでの許可の確認
- Microsoft Entra 管理センター

画面写真は転載していません。操作図はすべて説明用の模式図です。AADSTS9002326 の文言は、MicrosoftのQ&Aに記載されたプラットフォームのエラーメッセージに基づいています。