勤怠管理アプリ　導入手順書【公開版】

作成日：2026/09/29 元にした記録：テスト用会社（稼働テスト用）への導入、ライセンスの作り直し、位置情報の実機確認

> **公開版について**　この文書は、勤怠管理アプリを新しいお客さまに導入するときの、提供元（私）と客先の作業手順をまとめたものです。公開にあたり、提供元のメールアドレス、開発用パソコンのフォルダー名、ライセンス用の鍵の保管場所など、運用の内部情報だけを一般的な表現に置き換えています。手順の構成と内容は原文のままです。

この手順書の使い方

新しいお客さまに勤怠管理アプリを使い始めてもらうまでの手順です。上から順に進めてください。 各手順の頭に、**だれが行うか**を書いています。

呼び名	だれか
提供元	開発者（私）。パソコンの黒い画面（PowerShell）で命令を打つ作業も含みます
客先の管理用アカウント	お客さまの会社で、スプレッドシートの**持ち主**になるGoogleアカウント。**アプリはこのアカウントの権限で動きます**
客先の管理者	アプリにログインして初期設定をする人（会社の総務の方など）。管理用アカウントと同じ人のことが多いです

なぜ持ち主とデプロイを客先の管理用アカウントにするのか

2026/09/24の実験で、次のことが確かめられています。

- 提供元（編集者として招待される立場）は、**コードの反映と関数の実行はできますが、デプロイはできません**（「この操作を行う権限がありません」と出る）。
- デプロイできるのは持ち主だけなので、アプリは**必ず持ち主の権限で動きます**。社員へのメールも持ち主のアカウントから届きます。

そのため、次のように役割を分けます。

- 持ち主・デプロイ・定期処理の導入・シートの保護 … **客先の管理用アカウント**
- コードの反映・キーの発行 … **提供元**

全体の流れ

段階	内容	だれが
0	最初の1回だけの準備（ライセンス用の秘密鍵を作る）	提供元
1	スプレッドシートを作り、提供元を招待する	客先の管理用アカウント
2	コードを送り込む	提供元
3	シートを作る・最初の管理者を作る	提供元
4	デプロイする（アプリのURLを作る）	客先の管理用アカウント
5	定期処理を入れる・シートを保護する	客先の管理用アカウント
6	ライセンスキーを発行して登録する	提供元 → 客先の管理者
7	アプリの初期設定（会社情報・部署・社員など）	客先の管理者
8	社員への案内（ログインと位置情報の許可）	客先の管理者

---

0. 最初の1回だけの準備（提供元）

ライセンスキーの署名に使う**秘密鍵**を作ります。**勤怠管理アプリにつき1回だけ**です。お客さまごとには作りません。すでに作ってあれば、この段階は飛ばします。

1. パソコンで PowerShell を開き、勤怠管理アプリの開発フォルダーへ移ります。
2. 鍵を作る道具（`node tools/make-secret.js`）を実行します。
3. 「✅ 秘密鍵を作りました」と出れば完了です。鍵は開発用パソコンの、同期されない場所に保存されます（画面には鍵の文字は出ません）。
4. 鍵は**手書きのメモ**に写し、別の手段でも控えておきます。 - ⚠️ この鍵をなくすと、それまでに発行したライセンスキーが使えなくなります。 - ⚠️ 「秘密鍵のファイルはもうあります」と出たら、すでに作ってあります。上書きはされません。

---

1. スプレッドシートを作り、提供元を招待する（客先の管理用アカウント）

1. 客先の管理用アカウントでGoogleにログインし、**Googleドライブ**を開きます。
2. 左上の「**新規**」→「**Googleスプレッドシート**」→「**空白のスプレッドシート**」を押します。
3. 左上の「無題のスプレッドシート」を押し、名前を **`勤怠管理_（会社名）`** にします。
4. 上のメニューの「**拡張機能**」→「**Apps Script**」を押します。スクリプトエディタが新しいタブで開きます。 - ⚠️ この操作で、スプレッドシートにつながったスクリプトが作られます。**必ず1回開いてください**（開かないと、提供元がコードを送り込む先がありません）。
5. スクリプトエディタの左端に縦に並んだ絵のうち、**歯車の絵（プロジェクトの設定）**を押します。
6. 「ID」の欄にある「**スクリプト ID**」の「**コピー**」を押し、提供元へ送ります。
7. スプレッドシートのタブに戻り、ブラウザの上のアドレス欄を見ます。`https://docs.google.com/spreadsheets/d/` と `/edit` の間の長い文字列が**スプレッドシートのID**です。これも提供元へ送ります。
8. スプレッドシートの右上の「**共有**」を押し、提供元のメールアドレスを入れて、権限を「**編集者**」にして「**送信**」を押します。 - 「組織外の人とは共有できません」などと出た場合は、お客さまのGoogle Workspaceの

管理コンソールで、ドライブの共有設定を「組織外のユーザーとの共有を許可」にしてからやり直します。

---

2. コードを送り込む（提供元）

1. PowerShell で勤怠管理アプリの開発フォルダーへ移ります。
2. 点検の道具を2つ通します。どちらも「✅」で終わることを確かめます。 `node tools/check.js node tools/client-check.js`
3. 納品用のコードを組み立てます。 `<スクリプトID>` と `<スプレッドシートのID>` は、段階1で客先から受け取ったものに置き換えます。 `node tools/package.js --script=<スクリプトID> --parent=<スプレッドシートのID>` - 「✅ dist/ を作りました」と出れば完了です。 `dist` フォルダに、客先へ渡すコードができます。 - 自動テストのファイルは外され、ライセンスの秘密鍵は `dist` の中だけで本物に差し替わります。開発用のコードには触れません。
4. `dist` フォルダへ移り、コードを送り込みます。 `cd dist clasp push --force cd ..` - ⚠️ 必ず `dist` フォルダの中で実行します。 `dist` の中の宛先は客先、外の宛先は開発用です。

---

3. シートを作る・最初の管理者を作る（提供元）

スプレッドシートを開き、「**拡張機能**」→「**Apps Script**」でスクリプトエディタを開きます。

3-1. シートを作る

1. 左のファイルの一覧で `server/Config.gs` を押します。
2. 上の実行ボタンの右にある関数の選択欄で `initializeSpreadsheet_forSetup` を選び、「**実行**」を押します。
3. 初めて実行するときは「**承認が必要です**」と出ます。
4. 「**権限を確認**」を押します。
5. 自分のアカウントを選びます。
6. 「**このアプリは Google で確認されていません**」と出たら、左下の「**詳細**」を押し、下に出る「**（プロジェクト名）（安全ではないページ）に移動**」を押します。
7. 「**許可**」を押します。
8. 下の「**実行ログ**」に「実行完了」と出れば完了です。スプレッドシートに「従業員マスタ」「システム設定」などのシートができ、「シート1」は消えています。

3-2. 最初の管理者を作る

1. 左のファイルの一覧で `server/Employee.gs` を押します。
2. 関数の選択欄で `runBootstrapFirstAdmin_forSetup` を選び、「**実行**」を押します。
3. 実行ログに「**初回管理者の仮パスワード:** ○○○○」と出ます。この仮パスワードを控え、客先の管理者へ伝えます。 - ユーザーIDは `admin` です。最初のログインのときに、好きなユーザーIDに変えられます。 - ⚠️ 「既に従業員が登録されているため実行できません」と出たら、すでに作ってあります。

---

4. デプロイする（客先の管理用アカウント）

アプリのURLを作ります。**必ず客先の管理用アカウントで行います**（提供元はデプロイできません）。

1. スプレッドシートを開き、「**拡張機能**」→「**Apps Script**」でスクリプトエディタを開きます。
2. 右上の青い「**デプロイ**」→「**新しいデプロイ**」を押します。
3. 「種類の選択」の右の**歯車の絵**を押し、「**ウェブアプリ**」を選びます。
4. 次のように入れます。 - 説明：`初回`（何でもかまいません） - 次のユーザーとして実行：**自分**（客先の管理用アカウントのメールアドレスが出ます） - アクセスできるユーザー：**全員**
5. 「**デプロイ**」を押します。「アクセスを承認」と出たら、3-1 の3と同じ手順で許可します。
6. 「**ウェブアプリ**」の欄に出るURL（`https://script.google.com/macros/s/…/exec`）の「**コピー**」を押して控えます。**これが社員が使うアプリのURLです。**
7. 確かめ：ブラウザのシークレットウィンドウ（Googleにログインしていない状態）でこのURLを開き、勤怠管理のログイン画面が出ることを確かめます。 - Googleのログインを求められる／「アクセス権がありません」と出る場合は、お客さまのGoogle Workspaceの管理コンソールで、ウェブアプリを組織外（ログインしていない人）へ公開できるかを確かめます。 - 個人のGoogleアカウント（Gmail）で持つ場合は、画面の上に「このアプリケーションは Google Apps Script のユーザーによって作成されたものです」という帯が出ます。Googleの仕組みで、消すことはできません。

---

5. 定期処理を入れる・シートを保護する（客先の管理用アカウント）

必ず客先の管理用アカウントで行います。

- 定期処理は、入れた人の権限で動きます。提供元が入れると、バックアップが提供元のドライブにでき、保護したシートにも書き込めなくなります。
- シートの保護は、実行した人だけが編集できる形になります。提供元が実行すると、アプリ（持ち主の権限で動く）が書き込めなくなるため、持ち主でないと実行できないようにしてあります。

スクリプトエディタで、次の関数を上から順に実行します（実行のしかたは 3-1 と同じ）。

順	ファイル	関数	中身
1	`server/Backup.gs`	`installDailyBackupTrigger_forSetup`	毎日深夜2時のバックアップ（7日分を残す。スプレッドシートと同じ場所に「勤怠管理システム_バックアップ」フォルダができる）
2	`server/PunchLog.gs`	`installWeeklyWorkContentReminderTrigger_forSetup`	毎週月曜8時、業務内容の未記入を承認者へ知らせる
3	`server/WeeklyAttendanceConfirmation.gs`	`installWeeklyAttendanceConfirmationReminderTrigger_forSetup`	毎週月曜8時、週次勤怠確認

順	ファイル	関数	中身
			の未提出を知らせる
4	`server/Backup.gs`	`listInstalledTriggers_forSetup`	確かめ：実行ログに「**トリガーの導入状況（3/3導入済み）**」と出ること
5	`server/Backup.gs`	`runBackupNow_forSetup`	確かめ：その場でバックアップを1つ作る。実行ログにバックアップの一覧が出ること
6	`server/SheetProtection.gs`	`protectAppSheets_forSetup`	シートの保護。実行

順	ファイル	関数	中身
			ログに「シート○枚を保護しました」と出ること

- 4で「[未導入]」と出た行があれば、その行に書いてある関数を実行してから、もう一度4を実行します。
- 6のあとは、スプレッドシートのシートを手で編集できるのは持ち主だけになります（アプリは普通に書き込めます）。あとで手で直す必要が出たら、`server/SheetProtection.gs` の `unprotectAppSheets_forSetup` で外し、直したら6でかけ直します。

---

6. ライセンスキーを発行して登録する

6-1. キーを発行する（提供元）

1. PowerShell で勤怠管理アプリの開発フォルダーへ移ります。
2. キーを発行する道具を実行します。お客さまの会社名と、段階1で受け取ったスプレッドシートのIDを渡します。 `node tools/license.js --company "○○建設株式会社" --sheet <スプレッドシートのID>` - 通常の納品はこれで**正規版（期限なし）**になります。 - お試し（60日）のときは、最後に `--type trial` を足します。
3. 「■ ライセンスキー」の下に出た1行（`.` で2つに分かれた長い文字列）が**ライセンスキー**です。そのまま客先の管理者へ送ります。

6-2. キーを登録する（客先の管理者）

キーを登録するまでは、**閲覧と出力だけ**ができ、打刻・申請・設定の変更などは止まっています（ログインと社員の登録はできます）。**最初のログインのすぐあとに登録してください。**

1. パソコンで、段階4で控えたアプリのURLを開きます。
2. 「ユーザーID」に `admin`、「パスワード」に段階3-2の仮パスワードを入れ、「**ログイン**」を押します。
3. 「**初回ログイン設定**」の画面が出ます。 - 「ユーザーID（初期adminから変更する場合のみ入力・半角英数字/-/_）」：変えたい場合だけ入れます。 - 「新しいパスワード（半角8文字以上・記号も使えます）」と「新しいパスワード（確認）」：同じものを入れます。 - 「メールアドレス」：管理者のメールアドレスを入れます。 - 「**設定して開始する**」を押します。
4. 左のメニューの「**設定**」にマウスを乗せ、右に出た一覧のいちばん上の「**システムの状態**」を押します。
5. カード「**ライセンス**」の「**ライセンスキーの登録・更新**」の欄に、提供元から届いたキーを貼り付け、「**登録する**」を押します。
6. 「ライセンスキーを登録しました（正規版(期限なし)）」と出て、札が「**有効**」になれば完了です。左のメニューのいちばん下にも「ライセンス:正規版(期限なし)」と出ます。 - 「このキーは別のファイル向けに発行され

ています」と出た場合は、カードに出ている「**このファイルのID**」を提供元へ伝え、キーを発行し直してもらいます。

---

7. アプリの初期設定（客先の管理者）

パソコンでログインし、左のメニューから次の順で設定します。メニューは見出し（「設定」「有給」など）にマウスを乗せると、右に中の画面が出ます。

順	メニュー	画面	すること
1	設定	**会社情報**	会社名（表示用）、年度開始月、締め日、36協定の月次残業上限、1日の所定労働時間、始業・終業時刻などを入れ、「**保存する**」を押す。位置情報を使う場合は「**出勤時に位置情報を必須にする**」に印を付ける（段階8を参照）
2	設定	**休日カレンダー設定**	「休日の曜日パターン」を選び、会社独自の休み（夏季休暇など）は「会社カレンダー」の欄で「**追加**」する。「**保存する**」を押す
3	設定	**部署マスタ管理**	「**部署を追加**」で部署を作る
4	設定	**従業員管理**	「**従業員を追加**」で社員を登録する。登録すると**仮パスワード**が出るので控え、本人へ伝える（あとから一覧の「**仮パスワードを表示**」でも見られる。本人が最初のログインを済ませると見られなくなる）
5	設定	**部署マスタ管理**	部署の「**編集**」から、その部署の**承認者**（所長など）を追加する。承認者は4で登録した社員から選ぶので、4のあとに行う
6	設定	**給与データ出力設定**	給与ソフトへ渡すCSVの列を合わせ、「**保存する**」を押す
7	有給	**有給 初期データ入力**	導入時点の有給の残り日数を社員ごとに入れ、「**入力した内容を登録する**」を押す

- 承認は「その部署の承認者が**全員**承認して成立」です。承認者を増やしすぎると、承認待ちで止まりやすくなります。
- 社員への連絡メールの送り主は、客先の管理用アカウントになります。

---

8. 社員への案内（客先の管理者）

8-1. ログインのしかた

社員へは、次の3つを伝えます。

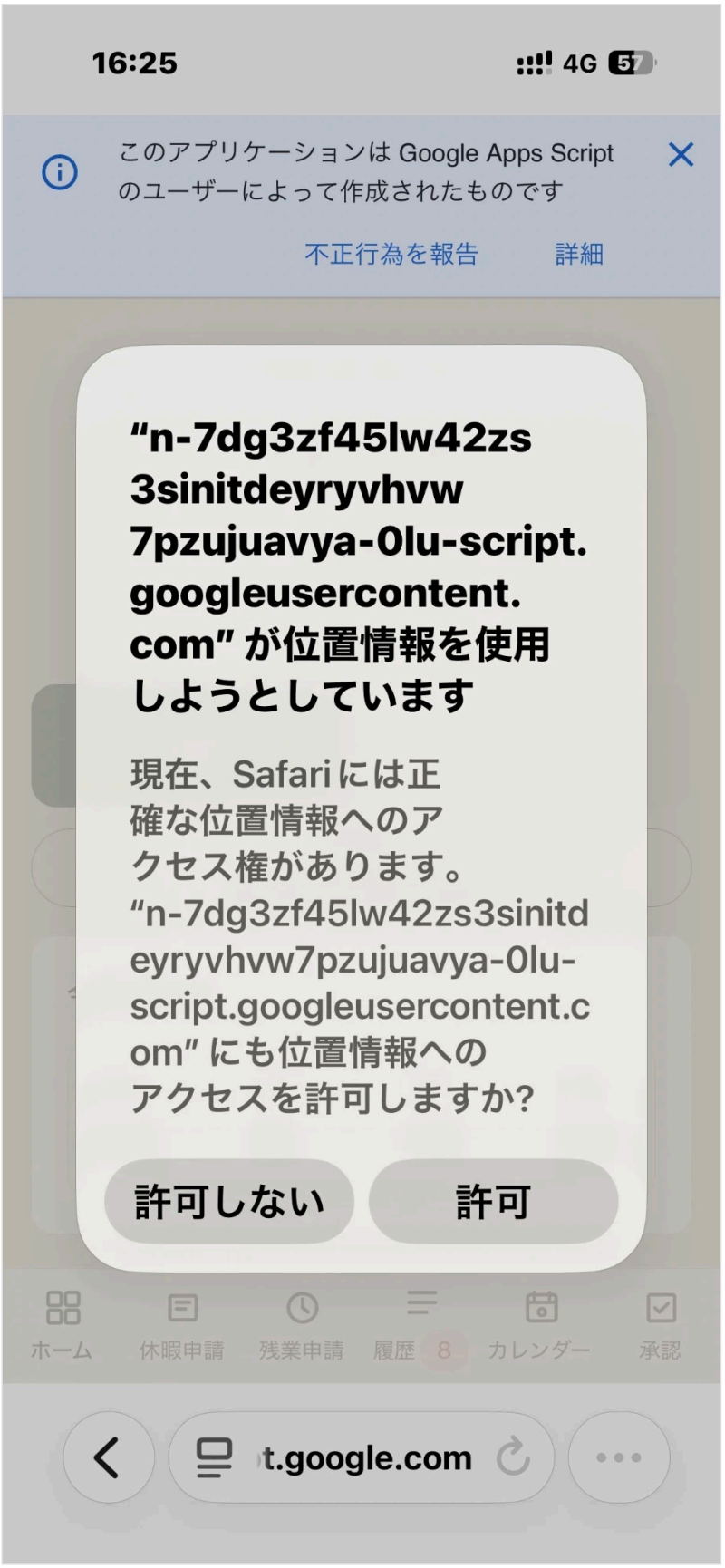
- アプリのURL（段階4で控えたもの）。スマホではホーム画面に追加しておくと便利です。
- ユーザーID
- 仮パスワード（最初のログインで、自分のパスワードとメールアドレスを設定します）

スマホでPC用の画面を開くと「スマホ・タブレットでアクセスしています」と出ます。そのときは「**モバイル版を開く**」を押してください。

8-2. 位置情報の確認画面（「出勤時に位置情報を必須にする」を使う場合）

- 位置情報を取るのは、**スマホ・タブレットで出勤・退勤を押したときだけ**です。パソコンでの打刻では取りません。
- 初めて出勤を押すと、スマホが位置情報の確認画面を出します。**「許可」を押してください。**
- 画面に出る名前は、Googleの仕組みのため、会社名ではなく次のようになります（変えられません）。社員が「怪しいサイト」と思って断らないよう、この写真を見せて案内してください。

Safari（iPhone）の場合 … 長い英数字の名前（`n-…-script.googleusercontent.com`）が出ます。



Chrome の場合 … `script.google.com` と出ます。



- **Androidで「正確な位置」と「おおよその位置」を選ぶ画面**が出たら、**「正確な位置」**を選んでください。「おおよその位置」だと数km単位でぼかされ、出勤したときに「位置が大まかにしか取れていません」と出ます（打刻は記録されます）。
- **一度「許可しない」を押してしまった場合**：ブラウザが覚えていて、確認画面が二度と出ません。iPhoneは「設定」アプリ →「アプリ」→「Safari」→「位置情報」を「確認」か「許可」にしてから、アプリの画面を

読み込み直してください。

- 「許可しない」のままだと、出勤は記録できません（退勤は記録できます）。どうしても位置情報を出せない日は、「打刻修正依頼」から申請してください。

---

9. 更新（バージョンアップ・修正）を届けるとき

保守契約のあるお客さまへ、修正を届ける手順です。

1. **（提供元）** 段階2の2〜4を行います（点検 → 納品用コードの組み立て → `dist` の中で `clasp push --force`）。
2. **（客先の管理用アカウント）** スクリプトエディタで、右上の「**デプロイ**」→「**デプロイを管理**」を押します。
3. 一覧の「**ウェブアプリ**」を選び、右上の**鉛筆の絵（編集）**を押します。
4. 「バージョン」を「**新バージョン**」にして、「**デプロイ**」を押します。 - ⚠️ 「新しいデプロイ」ではなく「デプロイを管理」から行います。新しいデプロイにすると**アプリのURLが変わり**、社員のホーム画面のリンクが使えなくなります。
5. アプリを開き直し、画面が普通に出ることを確かめます。

- ライセンスキーは、更新しても入れ直す必要はありません（同じ秘密鍵で組み立てているため）。
- 保守契約をやめたお客さまのソフトは、そのときのバージョンのまま使い続けられます。

---

10. 困ったとき

症状	原因と対処
打刻・申請などを押すと「ライセンスキーが登録されていません」と出る	段階6-2 がまだ。「設定」→「システムの状態」でキーを登録する
「このライセンスキーは別のファイル向けに発行されています」と出る	スプレッドシートをコピーしたか、バックアップから戻した。「システムの状態」の「このファイルのID」を提供元へ伝え、キーを発行し直す
「システムの状態」のシートの一覧に赤い行がある	見出しの数が違う・同じ名前の見出しが2つある・シートが無い。赤い行の名前と、そのシートの1行目（見出し）を提供元へ伝える（年度ごとのシートで、まだ使っていないものが「ありません」と赤くなるのは問題ない）
位置情報の確認画面が出ない	前に許可（または拒否）した内容をブラウザが覚えている。段階8-2 の「一度『許可しない』を押してしまった場合」と同じ場所で「確認」にすると、次から出る
バックアップができているか分からない	客先の管理用アカウントで `server/Backup.gs` の `listInstalledTriggers_forSetup` を実行し、「3/3 導入済み」かを見る。スプレッドシートと同じ場所の「勤怠管理システム_バックアップ」フォルダも見る