E6 Claude Code

APIキー（秘密）の扱い：.envに頼らず、Vaultに置く

.envが推奨されなくなってきた理由（根拠つき）、鍵をAIに読ませない設定、講師のVault方式、始め方、漏れたときの手順

更新日：2026-10-03（2026年10月時点の情報）｜HUMAC 伊藤晴通

対象ツール：Claude Code

考え方は、どのAIエージェントにも共通です。ただし、**ファイル名・画面・設定・コマンドは、Claude Code**（Claudeのデスクトップアプリの「Code」タブ）の場合です。ほかのAIエージェントでは、読み替えが必要です。

対象の方：開発者ではない方（経営者・事務・現場の担当者）。プログラミングは不要です。**コードは書かず、AIエージェントに頼んで、自分の業務アプリをつくる**ところまでを、順に進めます（ターミナルの黒い画面は、使わない形で書いています）。

結論：鍵は、AIが読めるファイルに置かない。置き場所は「金庫」

AIを使うアプリを作ると、「APIキー」（AIや外部サービスを使うための鍵）が必要になる。**鍵は、仕事のフォルダのファイルに、平文で書かない**。専用の「金庫」に入れ、**値は、AIの目に触れさせない**。講師は、**Supabase の Vault**に一元化している。この資料は、その考え方と、始め方。

1. 鍵が漏れると、何が起きるか（講師の実例）

講師は、AIを使うための鍵が公開の場所に出てしまい、**約6万8千円の不正請求**を受けた。鍵は、「財布の暗証番号つきのカード」のようなもの。**持っている人は誰でも、あなたの名前で、お金のかかる操作ができる**。そのため、鍵の扱いは、AI活用の中で最も慎重にする項目になる（→ E2）。

2. 「.env に鍵を書く」は、なぜ推奨されなくなってきたか

.env（ドット・エンブ）は、鍵などの設定を、1つのファイルにまとめて書く、長年の定番の方法。これまでは、「.gitignore で保存の対象から外せば安全」と言われてきた。しかし、AIエージェントの登場で、前提が変わった。

わかっていること	出典
AIエージェントは、作業フォルダの中のファイルを読む。**.env も、読む対象になりうる**。読んだ内容は、会話の文脈に入り、**モデルの提供会社のサーバーに送られる**	セキュリティ各社の指摘（Knostic、Infisical など）
2026年1月、「Claude Code が、`.claudeignore` や `.gitignore` で除外したはずの .env を読んでしまう」という報道があった。Anthropic はコメントしなかった	The Register（2026年1月28日）
Claude Code の公式ドキュメントは、読ませたくないファイルは、**設定の「Read の拒否ルール」**で指定すると書いている。そして、**.claudeignore は効果がない**と明記している	Claude Code 公式（権限）
拒否ルールが止められるのは、AIの標準の道具と、認識できるコマンドだけ。**AIが書いたスクリプトが、間接的にファイルを読む場合は止められない**。確実に止めるには、OSレベルの制限（サンドボックス）が要る	Claude Code 公式（権限）

「.env が禁止になった」のではない

正確には、**「AIが動く作業フォルダに、平文の鍵を置く運用」が、推奨されなくなってきた**。.env が必要な場面は、まだある（自分のパソコンで試すときなど）。だから、**「鍵を、AIの手の届かない所に置く」**のが、新しい基本になる。

3. 守りの3段階（鍵をどこに置くか）

段階	やること	向く場面
① 最低限	.gitignore で保存の対象から外す＋AIに**読ませない設定**（Read の拒否ルール）。使い終わったら、.env を消す	自分のパソコンで、ちょっと試すとき
② 鍵を、作業フォルダの外へ	使う場所に、直接しまう（Cloudflare の Secrets、Supabase の Edge Functions の Secrets）。保存後は、画面で値を読み出せない	小さなアプリが1つ、1か所で動くとき
③ 金庫に一元化（Vault）	すべての鍵を、1つの金庫に集める。アプリやサーバーは、起動時に、金庫から取り出す	複数のアプリ・パソコン・顧客で、鍵を使い回すとき。棚卸しと入れ替えを一元化したいとき

段階①：AIに、鍵のファイルを読ませない設定

仕事のフォルダの設定ファイル（`.claude/settings.json`）に、読ませないファイルを書く。ターミナルは使わず、Claudeに頼む。

コピーして使う

このフォルダの Claude Code の設定（.claude/settings.json）に、次のファイルを「読ませない」設定を追加してください：.env、.env.* 、secrets フォルダの中。
追加したら、実際に .env を読ませようとして、拒否されることを確認して、結果を見せてください（中身は表示しないでください）。

設定の書き方（公式）：`"permissions": { "deny": ["Read(.env)", "Read(.env.*)", "Read(secrets/**)"] }`。「拒否ルール」が効いているかは、必ず試して確認する。AIが書いたスクリプトが間接的に読む場合は止められないので、確実に守るなら、サンドボックス（公式）も併用する。

4. 講師の方式：Supabase の Vault に一元化する

Vaultは、Supabase に付いている「暗号化して鍵をしまう」機能。秘密は、暗号化されて保存される。復号するための鍵は、データベースの外の、保護された場所で管理される（公式）。講師は、**自社用も、顧客の業務用も、すべての鍵を、1つの金庫に集めて**いる。

1

1 人が、値を入れる
管理画面の、パスワード欄から。**チャットや、ファイルには書かない**

2

2 Vault に、暗号化して保存
名前と、用途の説明を付けて

3

3 取り出し口は、1か所
取り出すための関数を1つだけ作り、サーバーだけが呼べるようにする

4

4 アプリは、起動時に取り出す
メモリの中だけで使う。**ファイルに書かない**

設計のルール

- **値は、AIの目に触れさせない**。値の入力は、人が、管理画面でする。AIは「入れ物」と「取り出し口」を作るだけで、値は見ない。
- **ブラウザ・スマホのアプリから、直接取り出さない**。取り出せるのは、サーバー側（秘密の鍵を持つ処理）だけ。
- **取り出し口は、許可リスト方式**：「どのプログラムが、どの鍵を取れるか」を、決めておく。
- **名前の付け方を決める**：自分用は サービス名_用途（例：`GEMINI_API_KEY`）、顧客の業務用は `CLIENT_顧客の略称_サービス名_用途`。
- **説明（description）は必須**：用途・発行日・権限の範囲・誰宛か。
- **棚卸し**：半年ごとに、6か月以上更新していない鍵を確認する。使っていない鍵は、無効にする。
- **鍵ごとに、権限を最小にして発行する**：使い道だけに絞った鍵にする。

最後に残る、1本の鍵

金庫から取り出すためには、「金庫の鍵」が要る。**この1本だけは、どうしても、そのパソコン（サーバー）の環境に置く**（鶏と卵の問題）。講師は、これを、パソコンの暗号化（ディスク全体の暗号化）

で守り、**他の鍵は、すべて金庫に入れている**。環境に残る秘密が、1本だけなら、守りも棚卸しも、簡単になる。

金庫の中身を、AIに「見せて」と頼まない

金庫の鍵を、データベースの問い合わせ（SELECT）で表示させると、**画面に出た値が、会話の記録に残る**。講師は、これで実際に事故を起こし、鍵を入れ替えた。**確認してよいのは、「鍵が存在するか」「いつ更新したか」だけ**。値は見ない。試験には、使い捨てのダミーの鍵を使う。

5. 始め方（Claudeに頼む）

1

1 鍵の置き場所を作る

Supabase に、鍵専用のプロジェクトを1つ作る（試すだけなら無料でよい。業務で使うなら、下の注意を読む）

2

2 取り出し口を作る

Claudeに頼む。許可リスト方式で

3

3 ダミーで試す

使い捨ての値で、取り出せるか、権限が効くか確認

4

4 本物は、人が入れる

管理画面で。チャットに書かない

コピーして使う

Supabase の Vault を使って、APIキーを安全に管理する仕組みを作りたいです。次の条件で、設計を提案してください。作る前に、私の確認を取ってください。
・鍵の値は、私が管理画面で入れます。あなたは、値を見ません。チャットにも、ファイルにも、書かないでください。
・鍵を取り出す関数を1つだけ作り、サーバー側（service_role）だけが呼べるようにしてください。ブラウザ側の鍵（anon）からは、呼べないようにしてください。
・「どの呼び出し元が、どの鍵を取れるか」の許可リストを持たせてください。
・動作の確認は、使い捨てのダミーの鍵で行ってください。確認のとき、鍵の値は表示しないでください。
・名前の付け方は「サービス名_用途」。説明（description）を必須にしてください。

無料プランの「一時停止」に注意

Supabase の無料プロジェクトは、**1週間アクセスがないと、自動で一時停止**する（公式）。金庫が止まると、**アプリが起動時に鍵を取り出せず、動かなくなる**。毎日使うアプリが取りに来ていれば、止まらないが、**業務で使うなら、有料プラン（月25ドルから）を検討する**。無料のプロジェクトは2つまで（公式）なので、1つを鍵専用にすると、余裕が少なくなる点も、あらかじめ知っておく。

6. 日常の運用

場面	やること
新しい鍵を使い始める	管理画面で、名前と説明を付けて入れる。使うアプリの許可リストに足す
半年ごと	棚卸し。更新していない鍵、使っていない鍵を確認して、無効にする
鍵を入れ替える	発行元で新しい鍵を作り、金庫の値を置き換え、古い鍵を無効にする。アプリは、再起動して取り直す
鍵が漏れたかもしれない	①**すぐ、発行元でその鍵を無効にする** ②新しい鍵に入れ替える ③使われた形跡（請求・利用履歴）を確認する。**保存の履歴から消すだけでは足りない**（一度公開されたものは、取り戻せない）

7. 鍵の置き場所：どれを選ぶか

方法	向く場面	注意
.env（ファイル）	自分のパソコンで試すだけ	.gitignore ＋ AIに読ませない設定。使い終わったら消す。仕事のフォルダには、置きっぱなしにしない
Cloudflare Workers の Secrets	Cloudflare で動かす小さなアプリ1つ	保存後は、値を読み出せない（置き換えるだけ）。平文の環境変数には、鍵を置かない（公式）
Supabase Edge Functions の Secrets	Supabase の関数1つ	設定のとき、一時的に .env を使うことがある。使い終わったら消す。鍵の名前は SUPABASE_ で始められない（公式）
Supabase Vault（講師方式）	複数のアプリ・パソコン・顧客で、鍵を使い回す。棚卸し・入れ替えを一元化したい	取り出し口の設計が必要。無料の一時停止に注意

8. よくある失敗

鍵を、チャットに貼る

会話の記録に残り、提供会社にも送られる。値は、管理画面にだけ入れる。

.env を置きっぱなし

AIがいつでも読める状態になる。使い終わったら消すか、金庫に移す。

鍵を表示させて確認する

確認するのは「存在するか」だけ。値は見ない。

ブラウザ側に、秘密の鍵を入れる

画面のソースから、誰でも見られる。サーバー側だけで使う。

権限が広い鍵を使う

漏れたときの被害が大きい。使い道だけに絞った鍵にする。

入れ替えない・棚卸ししない

古い鍵が残り続ける。半年ごとに確認する。

9. 鍵を扱う前のチェックリスト

□	確認すること
□	鍵が、仕事のフォルダの、ファイル・指示書・チャットに書かれていない
□	.env を使う場合は、.gitignore と、AIに読ませない設定がある（試して確認した）
□	鍵の置き場所を決めた（試すだけ／Secrets／Vault）
□	値は、人が、管理画面で入れる。AIは、値を見ない
□	鍵の権限を、最小にしてある。名前と説明が付いている
□	ブラウザ・スマホ側には、秘密の鍵が無い
□	漏れたときの手順（無効化→入れ替え→確認）を知っている
□	半年ごとの棚卸しを、予定に入れた

関連資料

- E5 SupabaseとCloudflare … 動かす場所と、RLS・無料枠
- E2 ハーネスエンジニアリング入門 … 権限・関所と、事故から生まれたルール
- E4 Gitの導入と設定 … 鍵を保存の対象から外す（.gitignore）
- D1 信号機ルール

出典（2026年10月確認）

- Claude Code 公式：権限（Read の拒否ルール、.claudeignore は効果なし、サンドボックス） (https://code.claude.com/docs/en/permissions)
- Claude Code's prying AIs read off-limits secret files（The Register・2026年1月28日） (https://www.theregister.com/2026/01/28/claude_code_ai_secrets_files/)（報道）
- Your AI Coding Agent is Reading Your .env File（Infisical） (https://infisical.com/blog/your-ai-coding-agent-is-reading-your-env-file)（セキュリティ会社の解説）
- From .env to Leakage（Knostic） (https://www.knostic.ai/blog/claude-cursor-env-file-secret-leakage)（セキュリティ会社の解説）
- Supabase Docs：Vault (https://supabase.com/docs/guides/database/vault)
- Supabase Docs：Edge Functions の Secrets (https://supabase.com/docs/guides/functions/secrets)
- Cloudflare Workers：Secrets (https://developers.cloudflare.com/workers/configuration/secrets/)
- Supabase 料金（無料枠・一時停止） (https://supabase.com/pricing)