

個人用アクセス トークンを管理する

コマンド ラインまたは API を使って GitHub への認証を行うときに、パスワードの代わりに personal access token を使うことができます。

この記事の内容

personal access tokenについて

fine-grained personal access token

personal access token (classic) の作成

personal access token を削除する

コマンド ラインで personal access token を使用する

参考資料

警告: アクセス トークンは、パスワードと同様の扱いとしてください。詳しくは、「[personal access token のセキュリティ保護を維持する](#)」をご覧ください。

personal access tokenについて

[GitHub API](#) または [コマンド ライン](#) を使うときは、パスワードの代わりに Personal access token を使って、GitHub に対する認証を行うことができます。

Personal access tokenは、GitHub リソースに自動的にアクセスすることを目的としています。Organization に代わってリソースにアクセスするため、または有効期間の長い統合を行う場合、GitHub App を使用する必要があります。詳しくは、「[GitHub App の作成について](#)」を参照してください。

personal access token の種類

現在、GitHub では、2 種類の personal access token (fine-grained personal access token と personal access tokens (classic)) がサポートされています。GitHub では、可能な限り、personal access tokens (classic) ではなく fine-grained personal access token を使用することをお勧めします。

Organization の所有者は、personal access tokens (classic) のアクセス権を自分の Organization に限定するようにポリシーを設定できます。詳しくは、「[Organization の個人用アクセス トークン ポリシーを設定する](#)」を参照してください。

Fine-grained personal access token

Fine-grained personal access token には、personal access tokens (classic) に勝るいくつかの利点があります。

- 各トークンは、1 人のユーザーまたは 1 つの Organization が所有するリソースにのみアクセスできます。
- 各トークンは、特定のリポジトリにのみアクセスできます。
- 各トークンには特定のアクセス許可が付与されます。これにより、personal access tokens (classic) に付与されるスコープよりも細かく制御できます。
- 各トークンには、有効期限が必要です。
- Organization オーナーは、Organization 内のリソースにアクセスできる fine-grained personal access token の承認を要求できます。

Personal access tokens (classic)

Personal access tokens (classic) は安全性が低くなります。ただし、現在、一部の機能は personal access tokens (classic) でしか機能しません:

- 自分、または自分がメンバーではない組織によって所有されていないパブリック リポジトリに対する書き込みアクセス権を持つのは、personal access tokens (classic) のみです。
- 外部コラボレーターは、personal access tokens (classic) のみを使って、コラボレーターである組織のリポジトリにアクセスできます
- 一部の REST API 操作は、fine-grained personal access token では使用できません。fine-grained personal access token でサポートされている REST API 操作の一覧については、「[きめ細かい個人用アクセス トークンに使用できるエンドポイント](#)」を参照してください。

personal access token (classic) を使う場合は、それによって、そのユーザーがアクセスできる Organization 内のすべてのリポジトリと、そのユーザーの個人アカウント内のすべての個人リポジトリへのアクセスが許可されることに注意してください。

セキュリティ上の理由から、GitHub は過去 1 年間使われていない personal access token を自動的に削除します。セキュリティを強化するため、personal access token には有効期限を設けることを強くお勧めします。

personal access token のセキュリティ保護を維持する

Personal access token はパスワードのようなものであり、どちらにも同じ固有のセキュリティ リスクがあります。新しい personal access token を作成する前に、より安全な認証方法を使用できるかどうかを検討してください。

- コマンド ラインから GitHub にアクセスするには、personal access token を作成する代わりに、[GitHub CLI](#) または [Git Credential Manager](#) を使用できます。
- GitHub Actions ワークフローで personal access token を使う場合は、代わりに組み込み `GITHUB_TOKEN` を使用できるかどうかを検討してください。詳しくは、「[自動トークン認証](#)」を参照してください。

これらのオプションを使用できず、personal access token を作成する必要がある場合は、[1Password CLI](#) などの別のサービスを使ってトークンを安全に格納すること、または 1Password の [GitHub シェル プラグイン](#) を使って GitHub CLI に対する認証を安全に行うことを検討してください。

スクリプトで personal access token を使う場合は、トークンをシークレットとして格納し、GitHub Actions を使ってスクリプトを実行できます。詳しくは、「[暗号化されたシークレット](#)」をご覧ください。また、トークンを Codespaces シークレットとして格納し、Codespaces でスクリプトを実行することもできます。詳細については、「[codespaces の暗号化されたシークレットを管理する](#)」を参照してください。

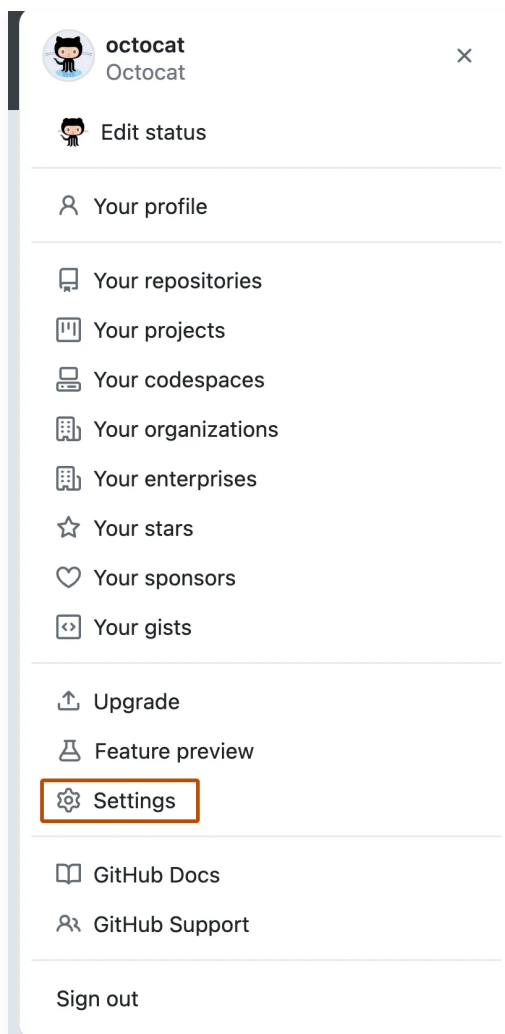
ベスト プラクティスについて詳しくは、「[API 資格情報をセキュリティで保護する](#)」をご覧ください。

fine-grained personal access token

の作成

注: Fine-grained personal access token は現在ベータ版であり、変更される可能性があります。フィードバックを残すには、[フィードバックのディスカッション](#)に関するページを参照してください。

- 1 まだ検証していない場合は、[メールアドレスを検証](#)します。1. 任意のページで、右上隅にあるプロフィールの画像をクリックし、次に[設定]をクリックします。



- 2 左側のサイドバーで **[<> 開発者設定]** をクリックします。
- 3 左側のサイドバーの **[🔑 Personal access token]** の下にある **[Fine-grained トークン]** をクリックしてください。
- 4 **[新しいトークンの生成]** をクリックします。
- 5 **[トークン名]** にトークンの名前を入力します。
- 6 **[有効期限]** で、トークンの有効期限を選びます。
- 7 必要に応じて、**[説明]** で、トークンの目的を説明するメモを追加します。
- 8 **[リソース所有者]** で、リソース所有者を選びます。トークンは、選んだリソース所有者が所有するリソースにのみアクセスできます。fine-grained personal access token にオプトインしない限り、所属している Organization は表示されません。詳しくは、「[Organization の個人用アクセス トークン ポリシーを設定する](https://docs.github.com/ja/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens)」をご覧ください。

- 9 必要に応じて、リソース所有者が **fine-grained personal access token** の承認を要求する Organization である場合、リソース所有者の下にあるボックスに、要求の正当な理由を入力します。
- 10 **[リポジトリ アクセス]** で、トークンでアクセスするリポジトリを選びます。ニーズを満たす最小限のリポジトリ アクセスを選ぶ必要があります。トークンには、GitHub 上のすべてのパブリック リポジトリへの読み取り専用アクセスが常に含まれています。
- 11 前の手順で **[リポジトリの選択のみ]** を選んだ場合は、**[選択されたリポジトリ]** ドロップダウンで、トークンでアクセスするリポジトリを選びます。
- 12 **[アクセス許可]** で、トークンに付与するアクセス許可を選びます。指定したリソース所有者とリポジトリ アクセスに応じて、リポジトリ、Organization、アカウントのアクセス許可があります。ニーズに必要な最小限のアクセス許可を選ぶ必要があります。各 REST API 操作に必要なアクセス許可について詳しくは、[「きめ細かい個人用アクセス トークンに必要なアクセス許可」](#) をご覧ください。
- 13 **[トークンの生成]** をクリックします。

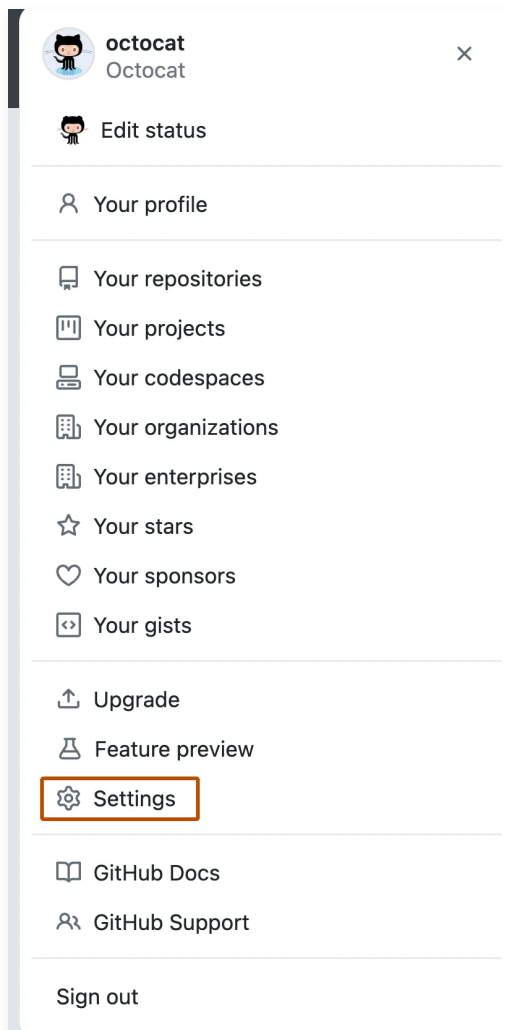
リソース所有者として Organization を選び、その Organization が **fine-grained personal access token** の承認を要求する場合、Organization 管理者によって確認されるまで、トークンは **pending** としてマークされます。トークンは、承認されるまでパブリック リソースの読み取りのみを実行できます。Organization のオーナーである場合、要求は自動的に承認されます。詳しくは、[「Organization での個人用アクセス トークンの確認と取り消し」](#) をご覧ください。


personal access token (classic) の作成

注: Organization オーナーは、personal access token (classic) のアクセスを自分の Organization に制限できます。personal access token (classic) を使用して、personal access token (classic) のアクセスが無効の Organization 内のリソースにアクセスしようとする、要求は 403 応答で失敗します。代わりに、GitHub App、OAuth app、または fine-grained personal access token を使用する必要があります。

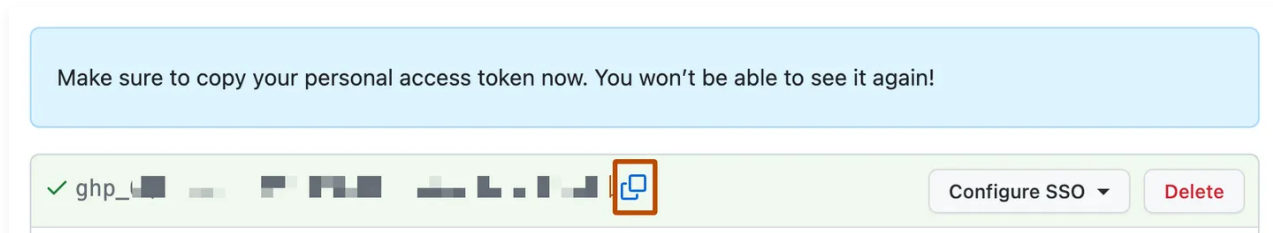
注: personal access token (classic) は、アクセスできるすべてのリポジトリにアクセスできます。GitHub では、代わりに、特定のリポジトリに制限できる fine-grained personal access token を使用することをお勧めします。Fine-grained personal access token を使用すると、広範なスコープの代わりにきめ細かなアクセス許可を指定することもできます。

- 1 まだ検証していない場合は、[メールアドレスを検証](#)します。1. 任意のページで、右上隅にあるプロフィールの画像をクリックし、次に[設定]をクリックします。



- 2 左側のサイドバーで [> 開発者設定] をクリックします。1. 左側のサイドバーの [ **Personal access token**] の下にある、[**トークン (クラシック)**] を選んでください。1. [**新しいトークンの生成**] を選び、[**新しいトークン (クラシック)**] をクリックします。
- 3 [メモ] フィールドで、トークンにわかりやすい名前を付けます。
- 4 トークンに有効期限を設定するには、[**有効期限**] を選び、既定オプションをクリックするか、[**カスタム**] をクリックして日付を入力します。
- 5 このトークンに付与するスコープを選びます。トークンを使用してコマンドラインからリポジトリにアクセスするには、[**リポジトリ**] を選択します。スコープが割り当てられていないトークンでは、パブリック情報にのみアクセスできます。詳細については、[「OAuth アプリのスコープ」](#)を参照してください。

- 6 **[トークンの生成]** をクリックします。
- 7 必要に応じて、新しいトークンをクリップボードにコピーするには、 をクリックします。

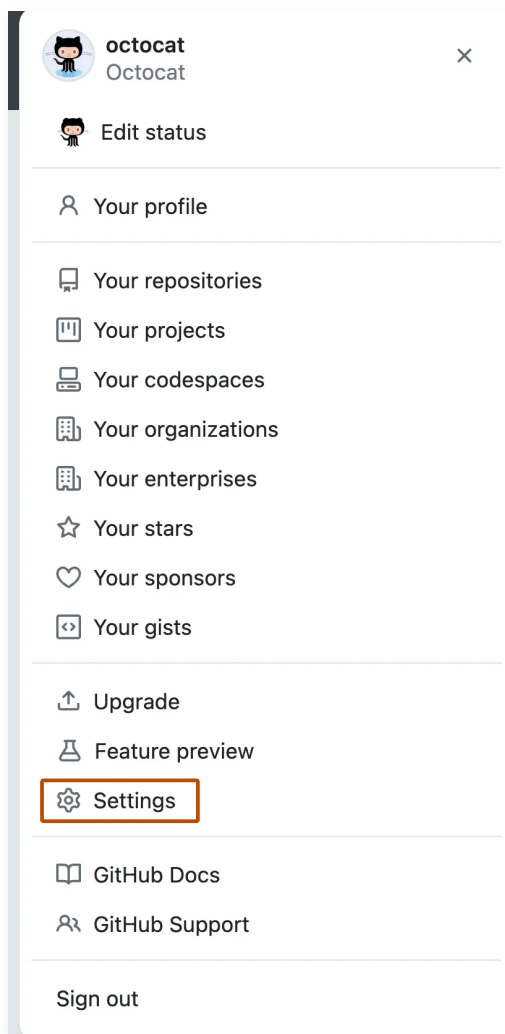


- 8 トークンを使用して、SAML シングル サインオンを使用する Organization が所有するリソースにアクセスするには、トークンを承認します。詳しくは、GitHub Enterprise Cloud のドキュメントの「[SAML シングルサインオンで利用するために個人アクセストークンを認可する](#)」をご覧ください。

personal access token を削除する

- 1 任意のページで、右上隅にあるプロフィールの画像をクリックし、次に**[設定]**をクリックします。





- 2 左側のサイドバーで [**<> 開発者設定**] をクリックします。
- 3 左側のサイドバーの  [**Personal access token**] で、削除する personal access token の種類に応じて、**[詳細なトークン]** または **[トークン (クラシック)]** をクリックします。
- 4 削除する personal access token の右側にある **[削除]** をクリックします。

コマンド ラインで personal access token を使用する

personal access token を入手したら、HTTPS 経由で Git の操作を実行するとき、パスワードの代わりにそれを入力できます。

たとえば、コマンド ラインでリポジトリをクローンするには、次の `git clone` コマンドを入力します。その後、ユーザー名とパスワードの入力を求められます。パスワードの入力を求められたら、パスワードの代わりに personal access token を入力します。


```
$ git clone https://github.com/USERNAME/REPO.git
Username: YOUR_USERNAME
Password: YOUR_PERSONAL_ACCESS_TOKEN
```

Personal access tokenは、HTTPS Git 操作にのみ使用できます。リポジトリで SSH リモート URL が使用されている場合は、[リモートを SSH から HTTPS に切り替える](#)必要があります。

ユーザ名とパスワードの入力を求められない場合、資格情報がコンピュータにキャッシュされている可能性があります。[キーチェーンの資格情報を更新して](#)、古いパスワードをトークンに置き換えることができます。

すべての HTTPS Git 操作でpersonal access tokenを手動で入力する代わりに、Git クライアントを使用してpersonal access tokenをキャッシュできます。Git では、有効期限が経過するまで、資格情報をメモリに一時的に格納します。また、Git ですべての要求の前に読み取ることができるプレーンテキスト ファイルにトークンを格納することもできます。詳しくは、「[Git に GitHub の認証情報をキャッシュする](#)」を参照してください。

参考資料

- [GitHub への認証方法について](#)
- [トークンの有効期限と取り消し](#)

法的情報

© 2023 GitHub, Inc. [用語](#) [プライバシー](#) [Status](#) [価格](#) [エキスパート サービス](#) [ブログ](#)