

Backlog 移行ツール (Zip版)

本移行ツールを用いて、課題やWiki、ドキュメントを別のスペースへ移行できます。同スペースでプロジェクトを分けたいときにもご利用できます。

必須要件

1. コマンドラインで簡単なコマンドを実行できる
2. 移行元のスペースの管理者、もしくは移行元プロジェクトのプロジェクト管理者であり、移行元のプロジェクトに参加している
[プロジェクト管理者の設定]
<https://support-ja.backlog.com/hc/ja/articles/360036147813>
3. 移行先のスペースの管理者である

任意要件

Backlog API の[課題一覧の取得]のクエリパラメータを作ることができる(移行する課題を絞り込みたいときのみ必要)

[課題一覧の取得]

<https://developer.nulab.com/ja/docs/backlog/api/2/get-issue-list/>

ダウンロード

以下から最新版の Zip ファイルを、お手元の環境にダウンロードしてください。

<https://backlog.com/ja/backlog-migration/releases.html>

Windows の場合

backlog-migration-<latest version>-windows.zip をダウンロードし、そのZipファイルを展開してください。

展開後、ターミナル内にて展開したフォルダまで移動してください。

Windows 11 の場合

展開したフォルダ内に移動します。

※展開時に使ったアプリケーションによって同じフォルダ名が入れ子になっていることがあるので、その際は再度同名フォルダ内へ移動します

次に、ウィンドウ内の何もないところで右クリックをするとメニューが展開されるので、その中から「ターミナルで開く」を実行します。

Windows 10 の場合

デスクトップ左下の検索バーをクリック、「cmd」と入力し、検索結果に表示される「コマンドプロンプト」をクリックして起動します。

上記展開したフォルダまで移動します。

binフォルダーにあるbacklog-migration.batを実行し、ヘルプが表示されるかお試しください。

macOS の場合

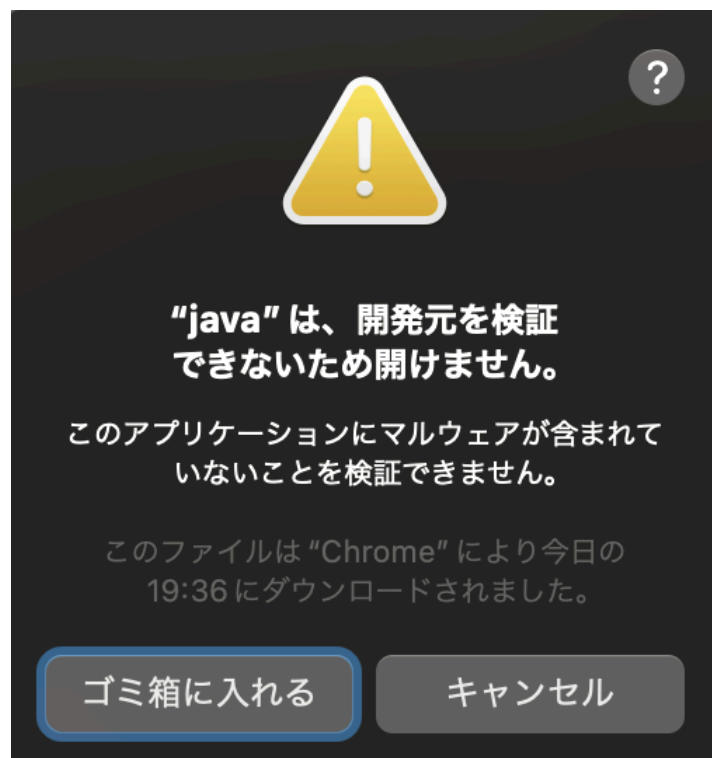
backlog-migration-<latest version>-macos-aarch64.zip をダウンロードし、そのZipファイルを展開してください。

展開後、Terminal.app ターミナル内にて展開したフォルダまで移動してください。
※展開時に使ったアプリケーションによって同じディレクトリ名が入れ子になっていることがあるので、その際は再度同名ディレクトリ内へ移動

ターミナルにて以下のコマンドでヘルプが表示されるかお試しください。

./bin/backlog-migration

上記コマンド実行時に下記のようなダイアログが表示される場合があります



この場合、下記を参考に macOS の設定を変更頂ければと思います。

<https://support.apple.com/ja-jp/guide/mac-help/mh40616/mac>

前準備

下記ページを参考に移行先と移行元 両方のスペースで、API キーを生成します。

[APIの設定]

<https://support-ja.backlog.com/hc/ja/articles/360035641754>

プロキシ設定を確認します。プロキシ経由で使用する場合はコマンドを以下のように指定します。

```
./bin/backlog-migration
-Djdk.http.auth.tunneling.disabledSchemes= \
-Dhttps.proxyHost=[プロキシサーバのホスト名/IPアドレス] \
-Dhttps.proxyPort=[ポート番号] \
-Dhttps.proxyUser=[プロキシユーザー名] \
-Dhttps.proxyPassword=[プロキシパスワード] \
...
```

移行

移行ツール実行時のトラブルシューティングを、[Backlogのプロジェクト移行ツールでエラーが発生して移行できません](#)にまとめています。移行の際にエラーメッセージが表示された場合は参照してください。

初期化処理

init コマンドを実行し、マッピングファイルを準備します。本移行ツールでは、移行元のユーザーを移行先に追加するのではなく、マッピングファイルを用いて移行元と移行先のユーザーを紐付けています。

ディレクトリ内でコマンドを以下のように指定します。

```
.\bin\backlog-migration \
init \
--src.key [移行元のAPIキー] \
--src.url [https://移行元のスペースID.backlog.jp] \
--dst.key [移行先の API キー] \
--dst.url [https://移行先のスペースID.backlog.jp] \
--projectKey [移行元のプロジェクトキー:移行先のプロジェクトキー]
```

以下のオプションは必須です。

--src.key	前準備で作成した移行元のAPIキー
--src.url	移行元のBacklog URL
--dst.key	前準備で作成した移行先のAPIキー
--dst.url	移行先のBacklog URL
--projectKey	移行元のプロジェクトキー、移行先のプロジェクトキーの順番でコロンを挟んで指定してください 例)--projectKey SRC_PRJCT:DST_PRJCT 移行元と移行先のプロジェクトキーが一致する場合以下のように省略可能です。 例)--projectKey PRJCT

init コマンドを実行すると、以下のようなディレクトリとファイルが生成されます。

実行ディレクトリ/

```
└ mapping/
  │   └ users.csv
  │   └ users_list.csv
  └ log/
    └ backlog-migration-warn.log
    └ backlog-migration.log
```

mapping/users.csv	ユーザーのマッピングファイルです。下記「マッピングファイルの修正」で使用します。
mapping/users_list.csv	設定可能なユーザーの一覧です。下記「マッピングファイルの修正」で使用します。
log/backlog-migration-warn.log	ログファイルです。init や execute 処理で致命的ではない警告ログが出力されます。
log/backlog-migration.log	ログファイルです。init や execute 処理の内容が追記出力されます。

マッピングファイルの修正

users.csvはcsv形式で出力されるテキストファイルです。**Destination Backlog user name**の項目を、users_list.csvの**Name**列から取得してください。すでに移行元のスペースから削除された等のユーザーアカウントである場合は、課題作成やコメント登録などを紐づける仮ユーザーを

移行先に追加してください。移行作業が終わった後、その仮ユーザーを削除しても問題ありません。

[ユーザーの追加・編集・削除]

<https://support-ja.backlog.com/hc/ja/articles/360035643534>

修正サンプル

users_list.csv ファイル

```
"Name","Email"
"user001","user001@example.com"
"user002","user002@example.com"
```

users.csv ファイル

```
"Source Backlog user id","Source Backlog user display name","Source Backlog user email","Destination Backlog user name"
"test001","テスト001","test001@test.com","user001"
"test002","テスト002","test002@test.com","user002"
```

移行

execute コマンドを実行すると、課題とWiki、ドキュメントがインポートされます。移行元が削除されることはありません。

ディレクトリ内でコマンドを以下のように指定します。

```
.\bin\backlog-migration \
execute \
--src.key [移行元のAPIキー] \
--src.url [https://移行元のスペースID.backlog.jp] \
--dst.key [移行先の API キー] \
--dst.url [https://移行先のスペースID.backlog.jp] \
--projectKey [移行元のプロジェクトキー:移行先のプロジェクトキー]
```

以下のオプションは必須です。

--src.key	前準備で作成した移行元のAPIキー
--src.url	移行元のBacklog URL
--dst.key	前準備で作成した移行先のAPIキー

--dst.url	移行先のBacklog URL
--projectKey	移行元のプロジェクトキー、移行先のプロジェクトキーの順番でコロンを挟んで指定してください 例)--projectKey SRC_PRJCT:DST_PRJCT 移行元と移行先のプロジェクトキーが一致する場合以下のように省略可能です。 例)--projectKey PRJCT

また、以下のオプションでカスタマイズできます。

--fitIssueKey	可能な限り課題番号を一致させます。課題詳細やコメント、Wiki に課題番号を記載している場合はこのオプションを指定することで、リンク切れを回避できます。移行先プロジェクトが既存のプロジェクトの場合は使用することはできません。
--importOnly	前回出力したファイルからインポートのみ実行します。
--exclude	課題やWiki、ドキュメントを移行しないよう制御できます。値は以下です。 --exclude wiki --exclude issue --exclude document
--retryCount	なんらかのエラーが発生したときの再試行回数を指定します。このパラメータを指定しなかったときの再試行回数は3回です。

移行する課題のフィルタリング

移行元のプロジェクトの課題をフィルタリングして、移行することができます。
フィルタリングの指定方法は、Backlog API v2の課題一覧の取得APIのクエリとほぼ同じです。

[課題一覧の取得]

<https://developer.nulab.com/ja/docs/backlog/api/2/get-issue-list/>

※ sort、order、projectId[]、カスタム属性(customField_)に関してはフィルタリングオプションとしてサポートしていません。

execute コマンドでオプション--filterを指定してください。(initコマンドでは使用できませんのでご注意ください。) 例として、比較的によく使われそうなフィルタリングオプションを以下の表に示します。詳細は上記の「課題一覧の取得」ページをご確認ください。

オプション指定例	型	内容
--filter "issueTypeId[]=1&issueTypeId[]=2"	数値	種別のIDでフィルタリングします。ID がわからない場合は、Backlog をブラウザで開き、プロジェクト設定>

		種別の一覧画面で、種別名にカーソルを合わせると EditIssueType.action?issueType.id=x とリンクが表示されます。issueType.id の値を指定してください。複数指定したい場合は、オプション指定例のように、複数の issueTypeId[] を & で連結してください。
--filter "categoryId[]=99&categoryId[]=98"	数値	カテゴリーのIDでフィルタリングします。ID がわからない場合は、Backlog をブラウザで開き、プロジェクト設定 > カテゴリーの一覧画面で、カテゴリー名にカーソルを合わせると EditComponent.action?component.id=x とリンクが表示されます component.id の値を指定してください。
--filter "id[]=123&id[]=456"	数値	課題のIDでフィルタリングします。移行に失敗した課題が明確な場合、その課題の移行のみをリトライしたい時に指定すると便利です。
--filter "offset=1000"	数値	課題のオフセットです。移行する課題が大量にある場合、時間を要するため、このオプションで分割して移行されることをお勧めします。 --filter count と組み合わせてご利用ください。 例) 1件目から100件移行したい場合 --filter "offset=0&count=100" 例) 1001件目から100件移行したい場合 --filter "offset=1000&count=100"
--filter "count=100"	数値	課題の移行最大数です。 --filter offset と組み合わせてご利用ください。

再移行時の重複チェック

「課題を登録したユーザーID」、「課題の登録時間」、「課題の件名」が全て一致する場合は、すでに移行された課題として、重複して移行されることはありません。何らかのエラーで移行が途中で終了し、再度 execute が必要なとき等に有益な重複チェックです。

「Wikiのページ名」が一致する場合は、すでに移行されたWikiとして、重複して移行されることはありません。再度 execute する際に有益な重複チェックです。

ドキュメントについては、名前と作成日時が全て一致する場合、すでに移行されたと見なして重複して移行されることはありません。

ログファイル

本移行ツールのディレクトリ構造は以下の通りです。ログファイルはlogディレクトリ内に生成されます。

```

実行ディレクトリ/
├ mapping/
└ └ users.csv

```

```
|   └─ users_list.csv
└─ log/
    └─ backlog-migration-warn.log
        └─ backlog-migration.log
```

制限事項

1. 親課題を変更後にプロジェクトキーを変えた場合、移行元では親課題の変更を反映できません。
2. 過去に存在した課題種別は移行中に自動作成されます。移行完了後、必要に応じて手動で削除してください。
3. 課題やwikiに付けられたスターは移行されません。
4. 数値のカスタム属性に単位を設定している場合、コメントの変更履歴にある単位は移行されません。
5. 移行元と移行先が同一プロジェクトは移行できません。
6. fitissuekeyとfilterオプションは同時に使用できません。
7. 移行元と移行先に同名のカスタム属性が存在する場合、移行先のデータに不整合が発生する可能性があります。実行する前に事前のチェックをお願いします。
8. グローバル検索のインデックスは移行されません。
9. プロジェクトに一度でも状態を追加した場合はASPからエンタープライズには移行できません。
10. 移行元もしくは移行先がフリープランの場合はAPIのレート制限により移行できません。
11. 本移行ツールの並列実行は、APIのレート制限を超える可能性があるため動作を保証できません。
12. 別のスペースに移行する際、Wiki の画像表示ダグをIDで表示している場合は画像は表示されません。
13. 「状態を追加」「状態名の変更」を9回以上おこなったプロジェクトは移行できません。
14. 種別、カテゴリ、状態、バージョン、マイルストーン、カスタム属性の名前の前後に空白スペースのある、または名前にカンマを含む場合は正しく移行できない可能性があります。移行の前に名前の前後にある空白スペース、または含まれているカンマを削除することをおすすめします。
15. 共有ファイル・Subversion・Gitは移行されません。
16. お知らせの既読状態は移行されません。

下位プランへの移行の際の注意点

下位プランへの移行の際、プロジェクト数、ユーザー数、添付できるファイルの数、添付できるファイルのサイズなどプランによる制限があります。Backlog移行ツールはこれらの制限を超えて移行することはできませんのでご注意ください。

[プランごとのファイルに関する制限](#)