AgileWorks R3

一括申請ガイド

R3.3 第 1 版(2026/07/15)

ATLED

目次／索引

1. 一括申請について 4
1.1. 一括申請の流れ 4
1.2. 一括申請ジョブの設定方法 4
1.3. 一括申請ジョブの実行方法 4
2. ジョブの設定 5
2.1. 一括申請ジョブ設定の新規作成 5
2.2. 一括申請ジョブ設定の確認と編集 7
2.3. 一括申請ジョブの通知設定 7
3. 書類データの準備 8
3.1. データ形式 8
3.2. CSV ファイルの設定例(表明細なし) 8
3.3. CSV ファイルの作成例(表明細あり) 9
4. 一括申請の実行 10
4.1. 管理サイトからの即時実行 10
4.2. 設定を用いたスケジュール実行 11
書類データファイルのアップロード 11
4.3. バッチ実行による即時実行 15
スクリプトの取得・構成・接続する AgileWorks の設定 15
スクリプトの実行 15
インポート結果の確認 17
5. 一括申請ジョブの結果確認 18
5.1. ステータスの確認 18
5.2. 実行状況・実行結果の詳細確認 19
6. 一括申請に関するエクスポート 20
6.1. 一括申請ジョブ設定のエクスポート 20
6.2. テンプレートファイルのエクスポート 20

◆ 改版履歴

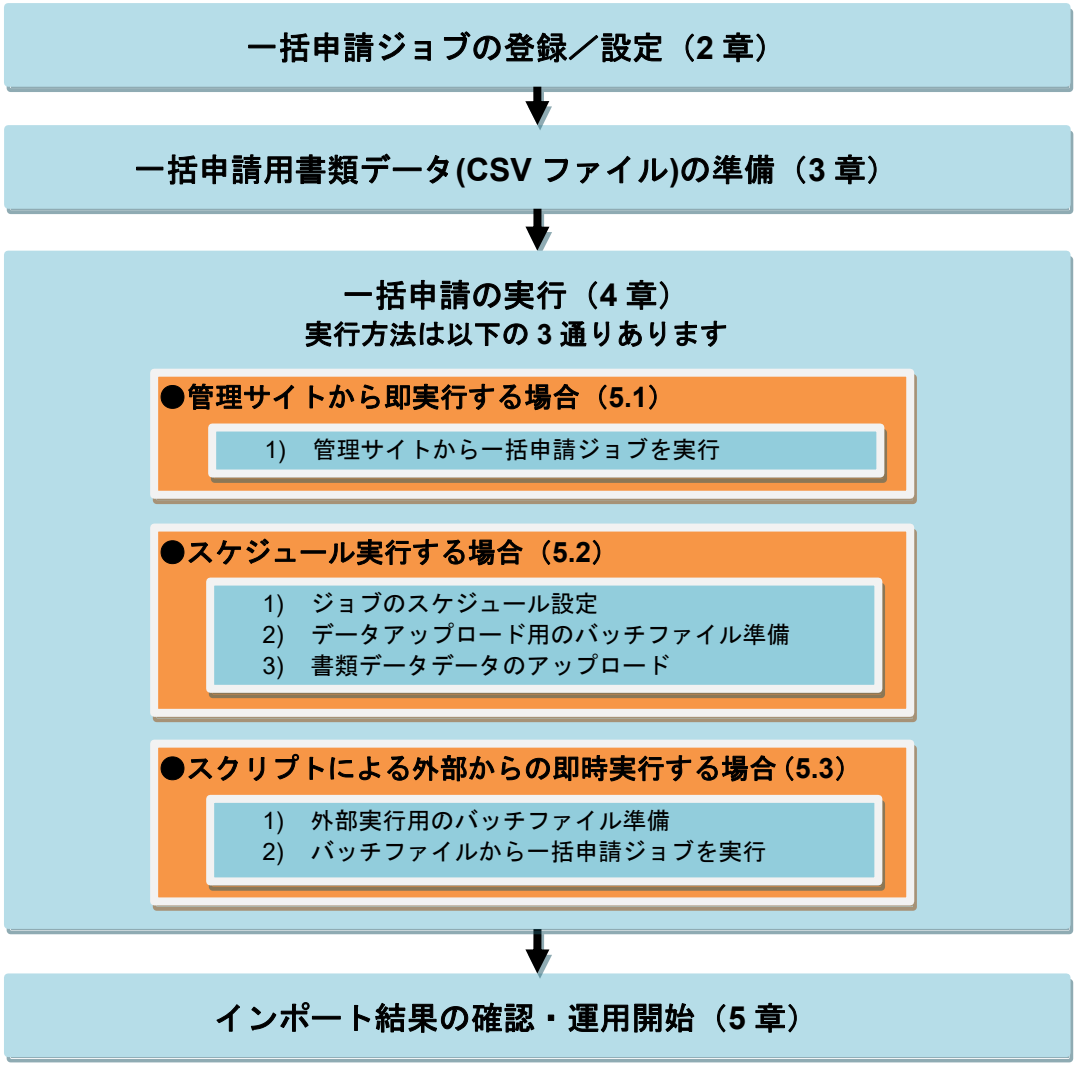
版数	年月日	改版内容
第 1 版	2026 年 07 月 15 日	第 1 版作成

1. 一括申請について

一括申請とは、AgileWorks のジョブ機能を利用して CSV ファイルで用意した書類データを一度に申請する機能です。
ジョブ設定を任意に追加し、設定されたジョブについて即時実行したり、スケジュール実行することができます。

1.1. 一括申請の流れ

一括申請は以下のような手順で設定・実行します。



1.2. 一括申請ジョブの設定方法

一括申請ジョブの設定は AgileWorks の管理サイトから行います。
具体的な設定については、「一括申請ジョブの設定」の章を参照してください。

1.3. 一括申請ジョブの実行方法

一括申請ジョブの実行は以下の 3 通りあります。
実行方法については、それぞれの章を参照してください。

- ジョブ設定画面から即時実行
- ジョブ設定画面からスケジュール設定して、定期的に実行
- バッチ実行による即時実行

2. ジョブの設定

一括申請のジョブ設定・実行の操作方法について説明します。

2.1. 一括申請ジョブ設定の新規作成

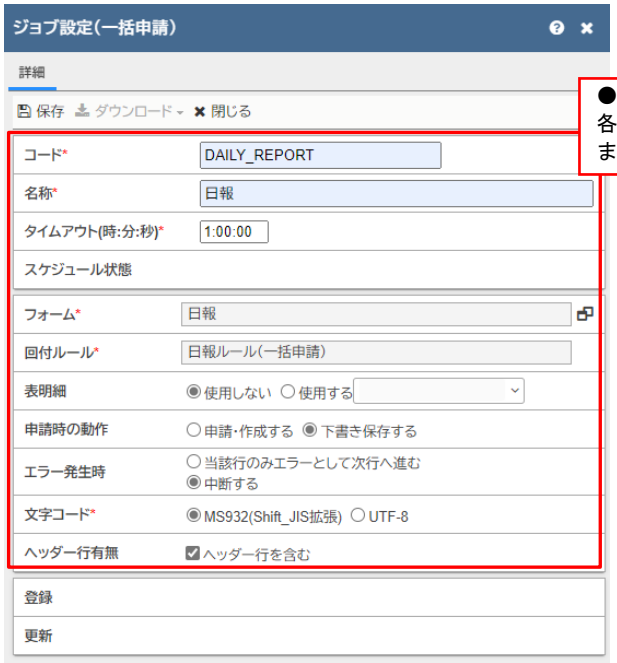
以下の画面からジョブ設定を新規作成します。

▼管理サイト【サイト管理】→【サイト共通設定】→【ジョブ／一括申請】



●管理サイト【サイト管理】→【サイト共通設定】→【ジョブ／一括申請】画面のメニューバーから【新規】→【ジョブ設定(一括申請)】をクリックします。

▼ジョブ設定[詳細]



●ジョブ設定の保存
各項目を入力後、【保存】を押下し、設定を保存します。



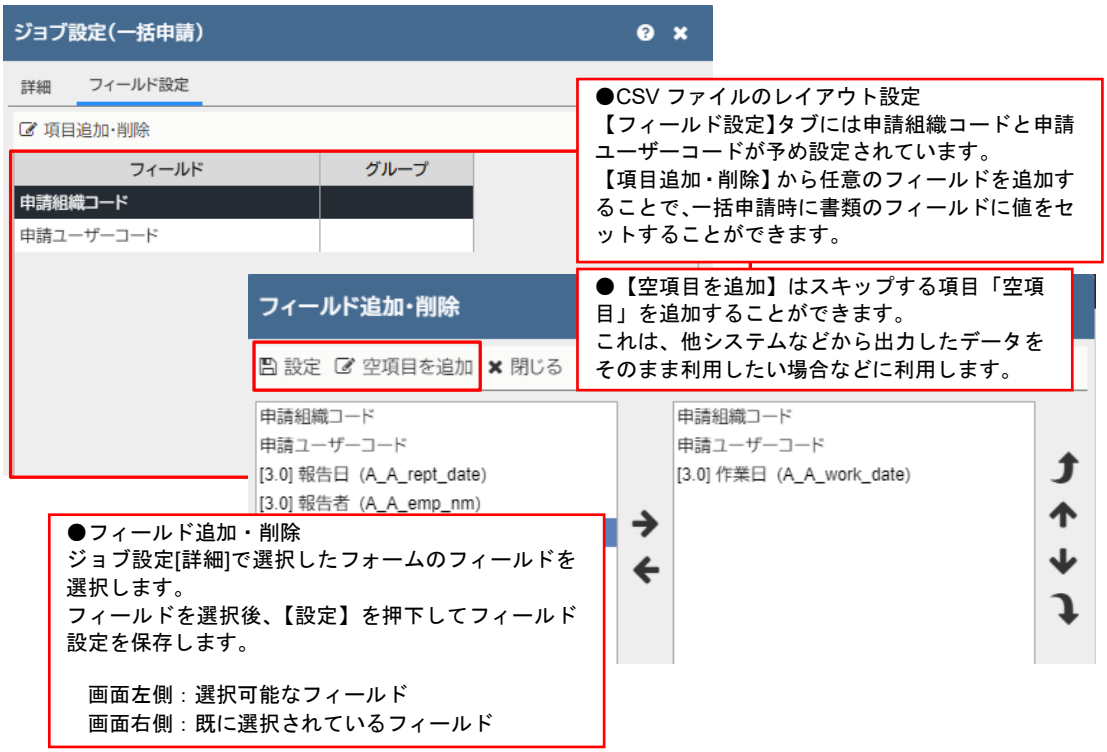
●CSV ファイルのレイアウト設定
一度設定を保存すると、【フィールド設定】タブが表示されます。
この【フィールド設定】タブにて、一括申請に利用する CSV データのレイアウトを設定します。

▼一括申請ジョブ[詳細]の設定項目

項目	説明
コード	一括申請ジョブを識別するコード。半角英数の文字列を指定します。 また、一括申請ジョブ内で一意である必要があります。 ※100 文字まで
名称	一括申請ジョブを識別する名称。日本語を指定できます。 ※255 文字まで
タイムアウト(時:分:秒)	一括申請ジョブがタイムアウトする時間を指定します。 タイムアウトするとジョブが中断されます。タイムアウトするまでに一括申請された書類は作成されたまま残ります。
フォーム	一括申請に利用するフォームを指定します。 ※公開フォームから回付ルールとセットで選択します。
回付ルール	一括申請に利用する回付ルールを指定します。 ※公開フォームからフォームとセットで選択します。
表明細	対象フォームの表明細に値をセットするかどうかを指定します。 ※一括申請で利用できる表明細は 1 つのみです。 複数の表明細を持ったフォームを利用する場合は、どれか 1 つにのみ用意 CSV から値をセットすることができます。 ※表明細を利用した書類データの作成方法については「書類データの準備」を参照ください。
申請時の動作	以下のどちらかを指定します。 ・申請・作成する ※対象の回付ルールに作成ステップが存在する場合は作成ステップにて【送信】 申請ステップしか存在しない場合は申請ステップにて【申請】が実行されます。 ・下書き保存する
エラー発生時	CSV ファイルに不正なデータが含まれていた際の動作を指定します。 ・当該行のみエラーとして次行へ進む ・中断する（エラー行以前の行は申請されます）
文字コード	CSV ファイルの文字コードを指定します。 ※MS932(Shift_JIS 拡張)か UTF-8 のいずれかを指定
ヘッダー行有無	CSV ファイルの 1 行目をヘッダー行として読み飛ばすか否かを指定します。

一括申請ジョブの実行時間について
一括申請ジョブの実行時間は CSV ファイルのデータ量や AgileWorks の利用状況などに影響されます。
大量の書類を一括申請する場合は、夜間にスケジュール実行するなどの運用をお願いします。
また、夜間に実行しユーザーが利用し始める時間にはジョブをタイムアウトさせるため、タイムアウト設定は最大 8 時間を目安に指定してください。

▼ジョブ設定[フィールド設定]

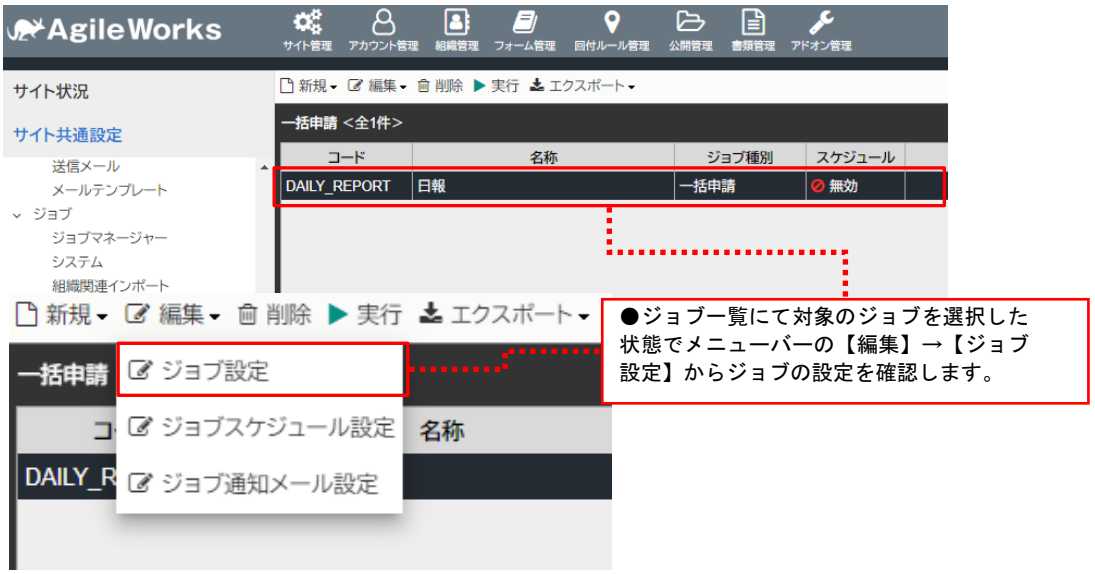


一括申請ジョブ[フィールド設定]で選択できるフィールドの詳細については「書類データの準備」を参照ください。

2.2.　一括申請ジョブ設定の確認と編集

保存したジョブは以下の画面にコード順で一覧されます。
この画面で、設定の変更やスケジュールの設定などを行うことができます。

▼管理サイト【サイト管理】→【サイト共通設定】→【ジョブ／一括申請】



2.3.　一括申請ジョブの通知設定

ジョブ通知メールを設定すると任意のタイミングで設定したメールアドレスにメールで通知することができます。
通知メール設定の詳細は別紙「ジョブガイド」の「ジョブの通知メール設定」を参照ください。

3. 書類データの準備

3.1. データ形式

以下の仕様で CSV ファイルを準備します。

▼CSV ファイルに関して

形式	説明
拡張子	可変長テキスト形式の CSV ファイル
文字コード	ジョブ設定にて指定した文字コードを使用してください。
区切り文字	カンマ「,」(半角) 書類に利用するデータ内にカンマが存在する場合は、項目の前後を半角のダブルクォーテーション「"」で囲む事によって利用できます。 ※カンマが存在しない場合でも囲んでも構いません。
囲み文字	ダブルクォーテーション「"」(半角) 書類に利用するデータ内に「"」が存在する場合は、項目の前後を半角の「"」で囲み、且つデータに利用したい「"」を 2 重に入力することで利用できます。 ※例）文字列「aaaa"bbbb」 > CSV データ上の入力内容「,"aaaa""bbbb",」
改行コード	「CR+LF／LF／CR」の 3 種類を改行として認識します。
1 行目の扱い	ジョブ設定の【ヘッダー行有無】の設定に従ってください。 ヘッダー行を含む場合、1 行目をスキップします。
フィールドの順番	ジョブ設定の【フィールド設定】の順番でデータを作成してください。 ※ヘッダー行の入力値と選択されているフィールド名が一致する必要はありません。
表明細を利用する場合	「CSV ファイルの作成例(表明細あり)」を参照ください。

▼データの入力値に関して

項目	必須	説明
申請組織コード		申請者、又は作成者とするユーザーの所属組織コードを指定します。 指定しない場合、自動的に主務が選択されます。
申請ユーザーコード	○	申請者、又は作成者とするユーザーのコードを指定します。
フォーム上のフィールド	△	対象のフィールドに入力したい値を指定します。 一括申請で書類を作成する場合でも必須フィールドの入力チェックは動作しますので、書類の作成・又は申請時に必須に設定されているフィールドが存在する場合、対象のフィールドの関しては値を指定してください。

上記の形式で 1 から CSV ファイルを作成することもできますが、既に上記の形式となっているテンプレートファイルをエクスポートすることもできます。
テンプレートファイルのエクスポートに関しては「テンプレートファイルのエクスポート」を参照ください。

3.2. CSV ファイルの設定例(表明細なし)

▼ジョブ設定

項目	設定値
表明細	利用しない
文字コード	MS932(Shift_JIS 拡張)
ヘッダー行有無	ヘッダー行を含む
フィールド	申請組織コード 申請ユーザーコード 作成日 備考

▼CSV データ例

```
申請組織コード,申請ユーザーコード,作業日,備考
EIGYO101,SSHIMIZU,2014/06/01,"10,000 を超える場合は別途経費精算書を提出してください。"
EIGYO102,KTAKAHASHI,2014/06/01, 10,000 を超える場合は別途経費精算書を提出してください。
,MHARA,2014/06/01,""
```

1 行目：ヘッダー行を含む設定のためスキップします。
2 行目：入力内容にカンマ「,」を含むため、備考の入力内容をダブルクォーテーションで囲んでいます。
3 行目：入力内容にカンマ「,」を含んでいますが、ダブルクォーテーションで囲んでいないためエラーとなります。

4 行目：申請組織コードを省略しているので、MHARA さんの主務組織で書類が作成されます。
　　備考も省略しているので、備考には何もセットされずに書類が作成されます。
　　また、カンマ「,」が含まれていない場合でもダブルクォーテーションで囲んでもエラーにはなりません。

3.3. CSV ファイルの作成例(表明細あり)

表明細を利用した一括申請を行う場合は、以下のように CSV ファイルを作成します。

▼表明細を利用する際の注意点
- 1 つの書類に対するデータ(行)は連続させる。
- 1 つの書類に対するデータの内、表明細以外のフィールドは全ての行を同じ内容にする。
- 表明細フィールドには上のデータ(行)からセットされる。

▼ジョブ設定

項目	設定値
表明細	利用する
文字コード	MS932(Shift_JIS 拡張)
ヘッダー行有無	ヘッダー行を含む
フィールド	申請組織コード 申請ユーザーコード 作業日 用途(表明細)

▼CSV データ例

```
申請組織コード,申請ユーザーコード,作業日,用途(表明細)
EIGYO101,SSHIMIZU,2014/06/01,交通費
EIGYO101,SSHIMIZU,2014/06/01,宿泊費
EIGYO102,KTAKAHASHI,2014/06/01,交通費
EIGYO102,KTAKAHASHI,2014/06/02,宿泊費
EIGYO101,SSHIMIZU,2014/06/01,交際費
```

上記の例では書類が 3 件作成されます。

1 行目　：ヘッダー行を含む設定のためスキップします。
2-3 行目：表明細の「用途」以外のフィールドが同じ値のため、1 つの書類として扱われます。
　　「用途」には上から「交際費」「宿泊費」の順でセットされます。
　→　2-3 行の 2 行で 1 件

3-4 行目：表明細の「用途」以外に作業日の値が異なっているため、別々の書類として扱われます。
　→　3 行目の 1 行で 1 件
　　4 行目の 1 行で 1 件

5 行目　：表明細の「用途」以外の値が 1-2 行目と同じですが、行が連続していないため別の書類として扱われます。
　→　5 行目の 1 行で 1 件

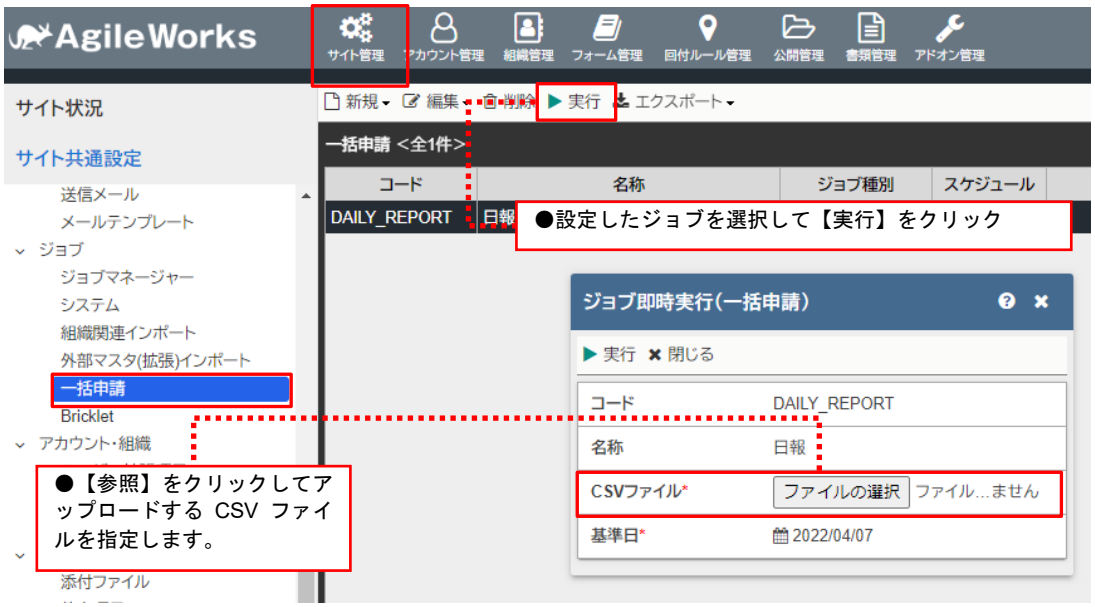
4. 一括申請の実行

この章では、外部マスタインポートを実行する 3 つの方法について説明します。

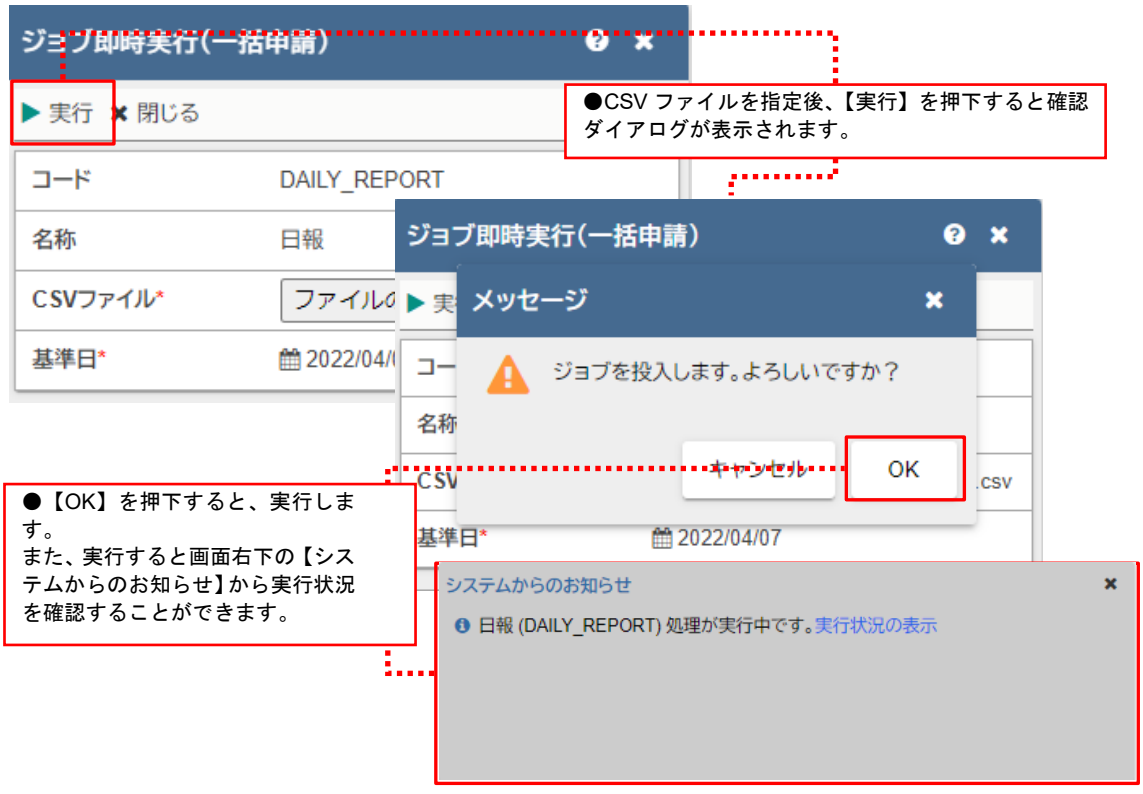
4.1. 管理サイトからの即時実行

以下の画面から書類一括申請ジョブを実行します。

▼管理サイト【サイト管理】→【サイト共通設定】→【ジョブ】



▼一括申請ジョブの即時実行

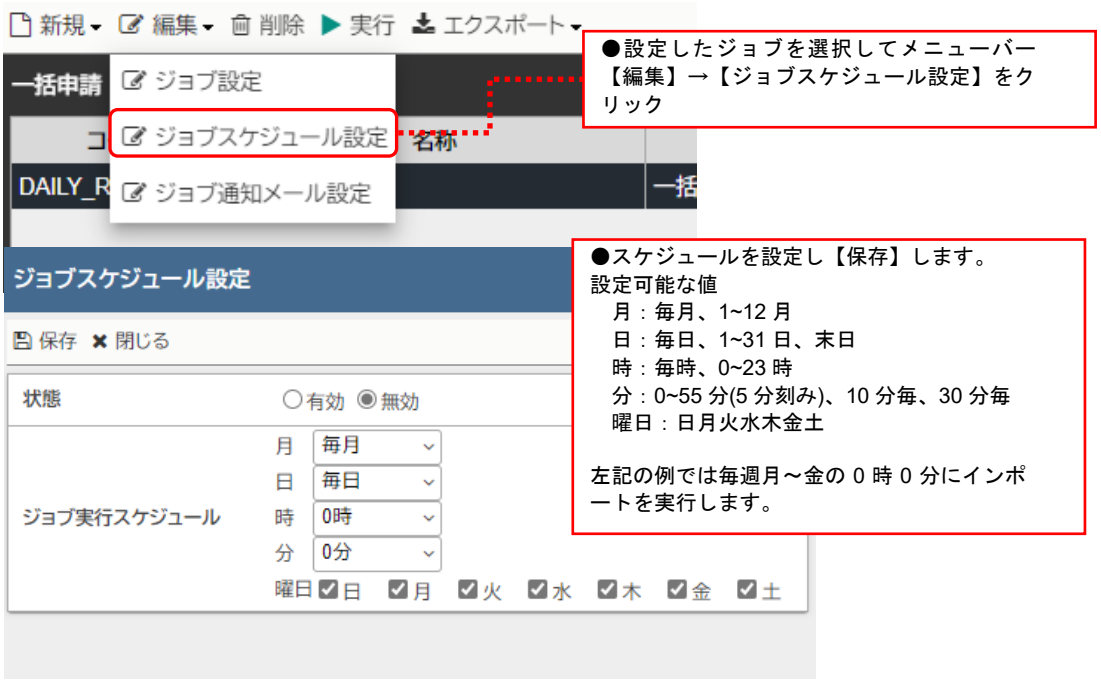


実行状況・実行結果の確認は「一括申請ジョブの結果確認」を参照ください。

4.2.　設定を用いたスケジュール実行

ジョブを定期的に実行するにはスケジュール設定を行ないます。

▼スケジュール設定（管理サイト【サイト管理】→【サイト共通設定】→【ジョブ／一括申請】）



スケジュールを設定すると、設定した時間帯に一括申請ジョブが実行されます。
一括申請ジョブで読み込む CSV ファイルは「書類データファイルのアップロード」の要領で、実行時間までに AgileWorks へアップロードしておきます。

書類データファイルのアップロード

インポートジョブをスケジュール実行する場合、実行時間までに、インポートするファイルをアップロードしておく必要があります。

スクリプトの実行条件

AgileWorks が提供するスクリプトは AgileWorks に対して、アクセス可能なネットワークポート(80 番など)でネットワーク接続が可能である環境において実行することができます。
また、スクリプトを実行する環境に Java7 以上の JRE がインストールされている必要があります。

【スクリプトの取得】

サポートサイトより zip ファイルをダウンロードし、スクリプトを実行する環境に配置・展開してください。
zip ファイルを展開すると、 remote というディレクトリが生成されます。
→　c:¥apps¥aw_bulk_apply¥remote(展開後のディレクトリ)　など

【スクリプトの構成】

remote ディレクトリ配下は以下の構成となっています。

remote 配下のディレクトリ	ファイル	備考
bin 配下	―	スクリプトの内部モジュール
bricklet 配下	―	Brickelt ジョブに利用するファイル ※書類ファイル出力では利用しない
bulk_apply	remote_cmd.bat	【実行スクリプト】 各スクリプトが内部で実行するスクリプト
	remote_cmd.sh	
	submit.bat	【ジョブ実行用スクリプト】 外部からジョブを実行する際に利用します
	submit.sh	
	submit_wait.bat	
	submit_wait.sh	
	upload.bat	【アップロードスクリプト】 書類データファイルのアップロードに利用します
	upload.sh	
conf	remote_env.bat	設定ファイル
	remote_env.sh	
doc_export 配下	―	書類ファイル出力に利用するファイル ※一括申請では利用しない
org_import 配下	―	組織インポートに利用するファイル ※一括申請では利用しない
user_mst_import 配下	―	外部マスタインポートに利用するファイル ※一括申請では利用しない

一括申請では上記の bin / conf / bulk_apply を利用します。

▼接続するAgileWorksの設定

conf/remote_env.sh(bat)を実行環境に合わせてテキスト編集します。

▼remote_env.bat(Windows)の場合

```
rem 接続先 URL を指定して下さい
set AWURL=http://XXX.co.jp/AgileWorks

rem ログイン ID を指定して下さい
set USER= bulk_apply_user

rem パスワードを指定して下さい
set PASS=********

・・・
```

●設定例

```
rem 接続先 URL を指定して下さい
set AWURL=https://atled.apply/AgileWorks

rem ログイン ID を指定して下さい
set USER=bulk_apply_user

rem パスワードを指定して下さい
set PASS=bulk_apply_pass
```

▼remote_env.sh(Linux)の場合

```
# 接続先サーバーを指定して下さい
AWURL=http:// XXX.co.jp /AgileWorks

# ログイン ID を指定して下さい
USER= bulk_apply_user

# パスワードを指定して下さい
PASS=********

・・・
```

【スクリプト内の編集】

設定値	説明
AWURL	接続先 AgileWorks の URL を指定します。
USER	実行ユーザーのログイン ID を指定します。 admin ユーザーを指定するか、または全コンテンツ・全業務カテゴリへの権限を持つ管理ロールを保持するユーザーを指定してください。
PASS	実行ユーザーのパスワードを指定します。

▼アップロードスクリプトの実行
bulk_apply/upload.sh(bat)
に引数を渡して実行します。

【スクリプトに渡す引数】
スクリプト実行時は以下の引数を指定してください。

引数名	必須	説明
CODE	○	対象の書類一括申請ジョブのコードを指定します。
DATA	○	アップロードするファイルをフルパスで指定します。
CRITERIONDATE		対象の申請基準日を YYYY-MM-DD 形式で指定します。 未指定の場合、システム日付で実行されます。 ※ステージング環境の場合はステージング基準日

【スクリプト実行方法（Windows）】
・ コマンドプロンプトを開きます。
・ 環境変数（JAVA_HOME）を設定します

```
set JAVA_HOME=c:¥PROGRA~1¥JAVA¥jdk1.8.0_60
```

※ この例では、C:¥Program File¥JAVA に JRE をインストールした環境を前提としています。
実際の設定値はお使いのマシン環境に合わせてください。

・ 実行ディレクトリへ移動します。

```
cd c:¥apps¥aw_bulk_apply¥remote¥bulk_apply
```

※ この例では、アップロードスクリプト一式を c:¥apps¥aw_bulk_apply に配置した環境を前提としているので、移動先ディレクトリはお使いのマシン環境に合わせてください。

・ ファイルアップロードを実行します。

```
upload.bat $CODE $DATA $CRITERIONDATE
```

【スクリプト実行方法（Linux）】
・ 環境変数（JAVA_HOME）を設定します

```
export JAVA_HOME=/usr/local/jdk1.8.0_60
```

※ この例では、/usr/local に JRE をインストールした環境を前提としています。
実際の設定値はお使いのマシン環境に合わせてください。

・ 実行ディレクトリへ移動します。

```
cd /home/aw_bulk_apply/remote/bulk_apply
```

※ この例では、アップロードスクリプト一式を/home/aw_bulk_apply に配置した環境を前提としているので、移動先ディレクトリはお使いのマシン環境に合わせてください。

・ 以下のコマンドを実行し、コピーしたスクリプトファイルに実行権限を付与します。

```
chmod a+x upload.sh
chmod a+x remote_cmd.sh
```

・ ファイルアップロードを実行します。

```
./upload.sh $CODE $DATA $CRITERIONDATE
```

アップロードスクリプトの実行例

【Windows 環境：スクリプトを c:¥apps¥aw_bulk_apply¥remote に配置した場合】

▼正常に実行できた場合

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>upload.bat DAILY_REPORT c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
  <Status>SUCCEED</Status>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

▼存在しないジョブコードを指定し、エラーとなった場合

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>upload.bat dairy_report c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
  <Status>FAIL</Status>
  <ErrorCode>APIJOB-W0001</ErrorCode>
  <HttpStatusCode>500</HttpStatusCode>
  <MessageText>APIJOB-W0001: 指定されたコード「daily_report」が登録されていません。</MessageText>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

【Linux 環境：スクリプトを/home/aw_bulk_apply/remote に配置した場合】

▼正常に実行できた場合

```
[root@localhost bulk_apply]# ./upload.sh DAILY_REPORT /home/data/daily_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>SUCCEED</Status>
</Response>
[root@localhost bulk_apply]#
```

▼存在しないジョブコードを指定し、エラーとなった場合(大文字小文字違い)

```
[root@localhost bulk_apply]# ./upload.sh daily_report /home/data/daily_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>FAIL</Status>
   <ErrorCode>APIJOB-W0001</ErrorCode>
   <HttpStatusCode>500</HttpStatusCode>
   <MessageText>APIJOB-W0001: 指定されたコード「daily_report」が登録されていません。</MessageText>
</Response>
[root@localhost bulk_apply]#
```

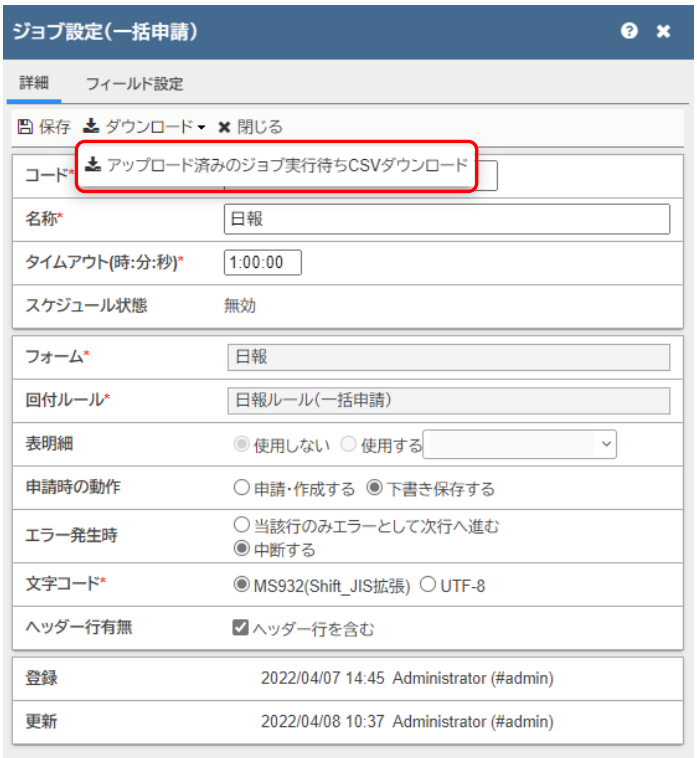
アップロードスクリプトのステータスコード

アップロードスクリプトを実行した際に、結果として表示される主なステータスコードです。

コード	説明
SUCCESS	アップロードが完了しました。
FAIL	アップロードに失敗しました。

アップロード結果の確認

【実行結果の確認】
管理サイト【サイト管理】→【サイト共通設定】→【ジョブ】→【一括申請】画面から、ファイルアップロードの実行結果を確認します。



このリンクが利用状態になっていればアップロードが成功したこととなります。
このジョブが実行されると、アップロードされたファイルは削除されます。

アップロードの注意事項
1 つの書類一括申請ジョブにつき、一度にアップロードできる CSV ファイルは 1 ファイルです。
アップロード後、スケジュール実行される前に再度 CSV ファイルをアップロードすると、最後にアップロードした CSV ファイルが有効となります。

4.3. バッチ実行による即時実行

AgileWorks が提供するスクリプトを利用することによって、AgileWorks が導入されている環境以外から書類一括申請ジョブを実行することができます。
このスクリプトはインポートする CSV のアップロードと、書類一括申請ジョブの実行を行います。

スクリプトの取得・構成・接続するAgileWorksの設定

インポートスクリプトはアップロードスクリプトと同じものを利用し、設定も同様となりますので、詳細は
「インポートファイルのアップロード」を参照してください。

スクリプトの実行

バッチ実行用のスクリプトは、2 種類用意されています。

ファイル	説明
bulk_apply /submit.sh(bat)	ジョブを実行します。実行結果は表示されません。
bulk_apply/submit_wait.sh(bat)	ジョブを実行し、完了するまで待機します。 ジョブの完了後、結果を表示します。

上記どちらかのスクリプトに以下の引数を渡して実行します。

【スクリプトに渡す引数】

引数名	必須	説明
CODE	○	対象の書類一括申請ジョブのコードを指定します。
DATA	○	アップロードするファイルをフルパスで指定します。
CRITERIONDATE		対象の申請基準日を YYYY-MM-DD 形式で指定します。 未指定の場合、システム日付で実行されます。 ※ステージング環境の場合はステージング基準日

【スクリプト実行方法（Windows）】

・ コマンドプロンプトを開きます。
・ 環境変数（JAVA_HOME）を設定します

```
set JAVA_HOME= c:¥PROGRA~1¥JAVA¥jdk1.8.0_60
```

※ この例では、C:¥Program File¥JAVA に JRE をインストールした環境を前提としています。
実際の設定値はお使いのマシン環境に合わせてください。

・ 実行ディレクトリへ移動します。

```
cd c:¥apps¥aw_bulk_apply¥remote¥bulk_apply
```

※ この例では、アップロードスクリプト一式を c:¥apps¥aw_bulk_apply に配置した環境を前提としているので、移動先ディレクトリはお使いのマシン環境に合わせてください。

・ 書類一括申請ジョブを実行します。

```
submit.bat $CODE $DATA $CRITERIONDATE
```

又は

```
submit_wait.bat $CODE $DATA $CRITERIONDATE
```

【スクリプト実行方法（Linux）】

・ 環境変数（JAVA_HOME）を設定します

```
export JAVA_HOME=/usr/local/jdk1.8.0_60
```

※ この例では、/usr/local に JRE をインストールした環境を前提としています。
実際の設定値はお使いのマシン環境に合わせてください。

・ 実行ディレクトリへ移動します。

```
cd /home/aw_bulk_apply/remote/bulk_apply
```

※ この例では、アップロードスクリプト一式を/home/aw_bulk_apply に配置した環境を前提としているので、移動先ディレクトリはお使いのマシン環境に合わせてください。

・ 以下のコマンドを実行し、コピーしたスクリプトファイルに実行権限を付与します。

```
chmod a+x submit.sh
chmod a+x submit_wait.sh
chmod a+x remote_cmd.sh
```

・ 書類一括申請ジョブを実行します。

```
./submit.sh $CODE $DATA $CRITERIONDATE
```

又は

```
./submit_wait.sh $CODE $DATA $CRITERIONDATE
```

バッチ実行スクリプトの実行例（Windows環境の場合）

スクリプトを c:¥apps¥aw_bulk_apply¥remote に配置した場合の例

【submit.bat の実行例】

・　正常に実行できた場合

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>submit.bat DAILY_REPORT c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>RUNNING</Status>
   <JobNo>000222</JobNo>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

・　存在しないジョブコードを指定し、エラーとなった場合(大文字小文字違い)

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>submit.bat daily_report c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>FAIL</Status>
   <ErrorCode>APIJOB-W0001</ErrorCode>
   <HttpStatusCode>500</HttpStatusCode>
   <MessageText>APIJOB-W0001: 指定されたコード「daily_report」が登録されていません。</MessageText>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

【submit_wait.bat の実行例】

・　正常に実行できた場合

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>submit.bat DAILY_REPORT c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>FINISHED</Status>
   <JobNo>000223</JobNo>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

・　申請ユーザーが指定されていないデータをインポートし、エラーとなった場合

```
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>submit_wait.bat DAILY_REPORT c:¥dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>ERROR</Status>
   <JobNo>000235</JobNo>
</Response>
c:¥apps¥aw_bulk_apply¥remote¥bulk_apply>
```

バッチ実行スクリプトの実行例（Linux環境の場合）

スクリプトを/home/aw_bulk_apply/remote に配置した場合の例

【submit.bat の実行例】

・　正常に実行できた場合

```
[root@localhost bulk_apply]# submit.sh DAILY_REPORT /home/data/dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>RUNNING</Status>
   <JobNo>000221</JobNo>
</Response>
[root@localhost bulk_apply]#
```

・　存在しないジョブコードを指定し、エラーとなった場合(大文字小文字違い)

```
[root@localhost bulk_apply]# submit.sh daily_report /home/data/dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>FAIL</Status>
   <ErrorCode>APIJOB-W0001</ErrorCode>
   <HttpStatusCode>500</HttpStatusCode>
   <MessageText>APIJOB-W0001: 指定されたコード「daily_report」が登録されていません。</MessageText>
</Response>
[root@localhost bulk_apply]#
```

【submit_wait.bat の実行例】

・　正常に実行できた場合

```
[root@localhost bulk_apply]# submit_wait.sh DAILY_REPORT /home/data/dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>FINISHED</Status>
   <JobNo>000236</JobNo>
</Response>
[root@localhost bulk_apply]#
```

・　申請ユーザーが指定されていないデータをインポートし、エラーとなった場合

```
[root@localhost bulk_apply]# submit_wait.sh DAILY_REPORT /home/data/dairy_report.csv 2017-03-01
<?xml version="1.0" ?>
<Response>
   <Status>ERROR</Status>
   <JobNo>000237</JobNo>
</Response>
[root@localhost bulk_apply]#
```


submit_wait.bat(sh)について
submit_wat.bat(sh)は実行したジョブが完了するまで画面には何も表示されません。
また、スクリプトの実行後、コマンドプロンプトを閉じたり、スクリプトの実行を中断するとジョブが実行されない場合があります。
従って、スクリプト実行後、ジョブの完了まで待機できる場合のみ利用してください。
例）夜間に submit_wait.sh を実行する仕組みを作成し、結果をログファイルに出力しておく　など

バッチ実行スクリプトのステータスコード

バッチ実行スクリプトを実行した際に、結果として表示される主なステータスコードです。

コード	説明
WAITING	待機中です。
CANCELED	ジョブを実行する前にキャンセルされました。
RUNNING	実行中です。
FINISHED	正常終了しました。
ERROR	エラーが発生しました。ログを確認してください。
INTERRUPTED	ジョブの実行中に中断されました
FAIL	ジョブの実行に失敗しました。 AP とジョブマネージャーの起動状態を確認してください。

インポート結果の確認

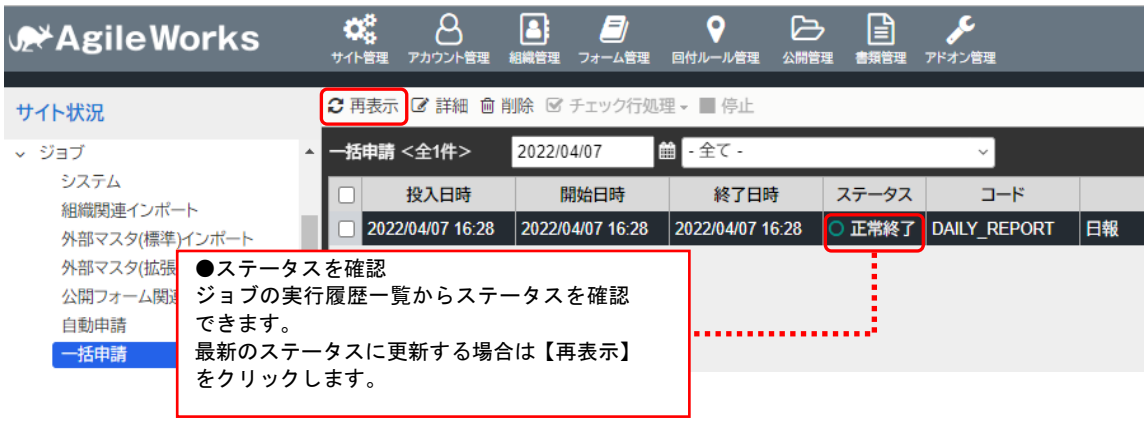
実行状況・実行結果の確認は「書類一括申請ジョブの結果確認」を参照ください。

5. 一括申請ジョブの結果確認

5.1. ステータスの確認

一括申請ジョブの履歴一覧から実行したジョブのステータスを確認することができます。

▼ステータスの確認（管理サイト【サイト管理】→【サイト状況】→【ジョブ／一括申請】）



▼ジョブのステータス

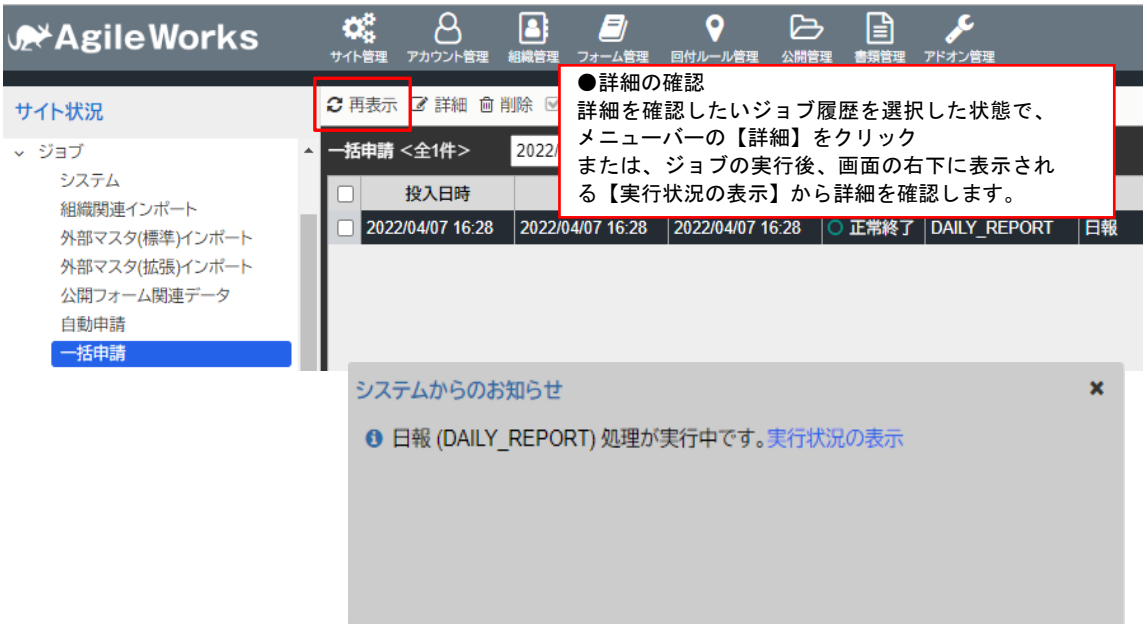
ステータス	説明
実行中	実行中です。 実際に処理されている、又は前のジョブの完了を待機している状態です。
正常終了	ジョブが正常に終了しました。
異常終了	ジョブの実行でエラーが発生しました。 【詳細】から状況を確認してください。
中断	何らかの原因でジョブが中断された状態です。 【詳細】から状況を確認してください。

ジョブの処理順序について
一括申請ジョブは実行された順番に処理されます。
既にジョブが実行されている状態で別のジョブを実行した場合は先に実行されているジョブの完了後、後に実行したジョブが処理されます。

5.2.　実行状況・実行結果の詳細確認

一括申請ジョブの実行状況・実行結果の詳細は以下のどちらかから確認します。
・　管理サイト【サイト管理】→【サイト状況】→【ジョブ／一括申請】
・　実行後に表示される【システムからのお知らせ】→【実行状況の表示】

▼管理サイト【サイト管理】→【サイト状況】→【ジョブ】→【一括申請】から実行状況と結果を核にします。



▼詳細の確認(正常終了)



6. 一括申請に関するエクスポート

一括申請ジョブに関する設定などをエクスポートすることができます。

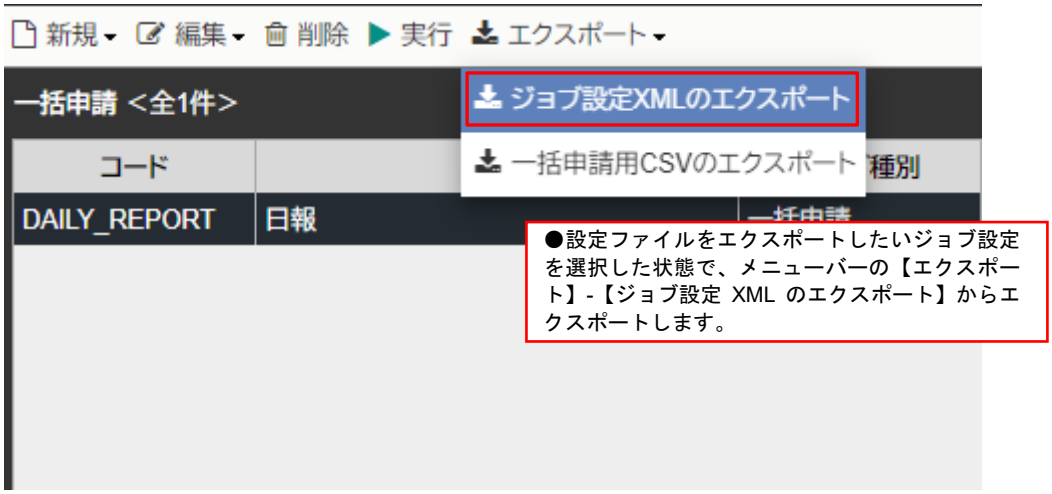
▼エクスポート可能な情報

情報	説明
ジョブ設定	一括申請ジョブ設定の XML ファイルです。 ジョブ設定のバックアップや他環境への移行時に利用できます。
書類データのテンプレートファイル	対象の一括申請ジョブに設定された項目がヘッダー行として入力されているテンプレートファイルです。

6.1. 一括申請ジョブ設定のエクスポート

ジョブ設定の XML ファイルをエクスポートします。

▼管理サイト【サイト管理】→【サイト共通設定】→【ジョブ】→【一括申請】にて対象のジョブを選択しエクスポートします。



6.2. テンプレートファイルのエクスポート

エクスポートしたテンプレートファイルを利用することで、一括申請用のデータファイルを作成する際に、ジョブの設定画面で必要な項目確認し、ヘッダー行を作成する作業を省くことができます。

▼管理サイト【サイト管理】→【サイト共通設定】→【ジョブ】→【一括申請】にて対象のジョブを選択しエクスポートします。

