



## **ASFALIS TransServer**

カスタマイズ機能開発向けAPIマニュアル

2020年 8月

株式会社エリジョン

# 目次

1. カスタマイズ機能開発向けAPI	1
1.1. 概要	1
1.2. システム要件	1
2. Web APIの利用手順	2
2.1. サーバ設定	2
2.2. Web API利用時の共通事項	2
2.3. ログイン	3
2.4. ログイン、ログアウト以外のAPIの利用	4
2.5. ログアウト	6
2.6. Web APIの処理結果	6
3. APIのパラメータ	11
3.1. ログイン(/login/authenticate)	11
3.2. オペレーション登録(/batch/job_lot_register)	12
3.3. オペレーション更新(/batch/job_lot_update)	17
3.4. ジョブ登録(/batch/job_entry)	22
3.5. ジョブステータス取得(/batch/job_state)	32
3.6. ジョブダウンロード(/batch/job_download)	33
3.7. ジョブダウンロード(直接) (/batch/job_download_direct)	36
3.8. ジョブ中断(/batch/job_stop)	40
3.9. ファイルアップロード(/batch/upload)	41
3.10. ファイルセット登録(/batch/file_set_register)	49
3.11. ファイル形式情報取得(/batch/mime_type_info)	53
3.12. ファイル型オペレーション属性ダウンロード (/batch/job_lot_attributes_download)	55
3.13. ファイル型オペレーション属性ダウンロード(直接) (/batch/job_lot_attributes_download_direct)	56
3.14. ユーザ情報取得(/batch/users)	58
3.15. グループ情報取得(/batch/groups)	60
3.16. ログアウト(/logout)	61
4. サンプルプログラム	62
4.1. ライセンスと免責条項	62
4.2. EntryAndDownload	62

# 1. カスタマイズ機能開発向けAPI

ASFALIS TransServer では、カスタマイズ開発をサポートするためASFALIS TransServer Web API (以下 Web API)を提供しています。

## 1.1. 概要

Web APIでは以下の機能を提供しています。

機能名	URL	概要
ログイン	/login/authenticate	ASFALIS TransServerにログインする。
オペレーション登録	/batch/job_lot_register	オペレーションを登録する。 オペレーション属性を登録する。 オペレーションに対する権限を登録する。
オペレーション更新	/batch/job_lot_update	オペレーション名を更新する。 オペレーション属性を更新する。 オペレーションに対する権限を更新する。
ジョブ登録	/batch/job_entry	ジョブを登録する。 変換パラメータを設定する。
ジョブステータス取得	/batch/job_state	指定ジョブのステータスを取得する。 ジョブを指定しない場合、ログインユーザが登録したジョブのステータスを全て取得する。
ジョブダウンロード	/batch/job_download	ジョブの変換結果ファイル、ログファイルをダウンロードする。
ジョブダウンロード (直接)	/batch/job_download_direct	ジョブの変換結果ファイル、ログファイルをHTTPで直接ダウンロードする。
ジョブ中断	/batch/job_stop	未処理または処理中のジョブを強制終了する。
ファイルアップロード	/batch/upload	ファイルセット登録に利用するファイルをアップロードする。
ファイルセット登録	/batch/file_set_register	ファイルセットを登録する。
ファイル形式情報取得	/batch/mime_type_info	指定シナリオまたはジョブの入力ファイル形式についての情報を取得する。
ログアウト	/logout	ASFALIS TransServerからログアウトする。

## 1.2. システム要件

Web APIを使用するには以下のライセンスが必要です。

※・ASF-API/S

## 2. Web APIの利用手順

### 2.1. サーバ設定

Web APIを使用するには、ASFALIS Trans Serverの[管理者機能]-[サーバ設定]の以下のパラメータに、Web APIを利用するコンピュータのIPアドレスを設定する必要があります。

- batch.client.address

### 2.2. Web API利用時の共通事項

- HTTPリクエストには、下記のHTTPヘッダを設定してください。  
X-ASFALIS-TRANSSERVER-BATCH: TRUE
- ログイン以外のHTTPリクエストには、認証情報を設定して送信してください。認証情報として、セッション情報と認証トークンを利用する方法と、アクセストークンを利用する方法があります。
  - セッション情報と認証トークンを利用する場合、ログインでSet-Cookieに設定されるセッション情報をCookieとして送信してください。  
セッション情報のキー名 `_K2Dashboard_session`
  - アクセストークンを利用する場合、アクセストークンを設定したユーザとしてログイン・ログアウト以外のAPIを利用できます。アクセストークンは、ユーザ編集画面から生成可能です。リクエストに付与するパラメータにこの値を`access_token`として追加して送信してください。  
(例)「[ログイン、ログアウト以外のAPIの利用](#)」の例をご参照ください。
    - [\[anchor3-9\]](#) については、アクセストークンを利用することができません。セッション情報と認証トークンを利用してください。
- ログイン以外のHTTPリクエストには、以下のContent-Typeを設定してJSON形式で指定されたパラメータを送信してください。  
Content-Type: application/json; charset=utf-8
- Web APIからのHTTPレスポンスはJSON形式で返却されます。  
ジョブダウンロード（直接）が正常終了するケースでは、HTTPレスポンスはダウンロードされるファイルの中身そのものになります。

## 2.3. ログイン

Web APIのログイン(/login/authenticate)にアカウント情報を送信します。リクエストには、下記のパラメータを付与してください。

- ログイン名、パスワード

認証に成功すると認証トークンが返却され、Cookieのセッション情報が設定されます。

(例)ログイン(POST /login/authenticate)

- リクエスト

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Content-Type: application/json; charset=utf-8

{
  "login_name": "admin",
  "password": "admin"
}
```

- レスポンス

```
Set-Cookie: _K2Dashboard_session=***** (セッション情報)
Content-Type: application/json

{
  "result_code": "0000",
  "authenticity_token": "*****" (認証トークン)
}
```

HeaderのSet-Cookieに以下のようにセッション情報が設定されます。ログイン以外のHTTPリクエストには、このセッション情報をCookieとして送信してください。

```
Set-Cookie: _K2Dashboard_session=***** (セッション情報); path=/; expires=Fri, 1-Jan-2012 12:00:00 GMT; HttpOnly
```

## 2.4. ログイン、ログアウト以外のAPIの利用

対象のWeb APIに下記のリクエストを送信します。

- セッション情報と認証トークン、またはアクセストークン
  - [\[anchor3-9\]](#) に限りアクセストークンを利用することはできません。
- API個別のパラメータ

API個別のパラメータについては、「3. API のパラメータ」を参照してください。

(例) ジョブステータスを取得(POST /batch/job\_state)

- リクエスト
  - セッション情報と認証トークンを送信する場合

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Content-Type: application/json; charset=utf-8

{
  "job_id": 1,
  "authenticity_token": "*****" (認証トークン)
}
```

- アクセストークンを送信する場合

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Content-Type: application/json; charset=utf-8

{
  "job_id": 1,
  "access_token": "*****" (アクセストークン)
}
```

- レスポンス

Content-Type: application/json

```
{
  "result_code": "0000",
  "jobs":
  [
    {
      "status": 3,
      "input_filename": "SAMPLE1.x_t",
      "created_at": "2012-10-04 10:50:00Z",
      "licence_busy": false,
      "id": 1
    }
  ]
}
```

## 2.5. ログアウト

Web APIのログアウト(/logout)に下記のリクエストを送信します。

- 認証トークン

(例) ログアウト(POST /logout)

- リクエスト

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Content-Type: application/json; charset=utf-8

{
  "authenticity_token": "*****" (認証トークン)
}
```

- レスポンス

```
Content-Type: application/json

{
  "result_code": "0000",
}
```

## 2.6. Web APIの処理結果

Web APIが返却する処理結果コード(result\_code)の内容は以下の通りです。

3000台： バッチ共通

4000台： ASFALIS TransServer共通

4100台： ASFALIS TransServer - JobLotRegister

4200台： ASFALIS TransServer - JobEntry

4300台： ASFALIS TransServer - JobState

4400台： ASFALIS TransServer - JobDownload

4500台： ASFALIS TransServer - JobStop

コード	説明	備考
0000	コマンド正常終了	
0001	コマンドオプション不正	必須項目が設定されていない 設定値の書式不正



コード	説明	備考
0002	ファイルエラー（対象が存在しない、ファイルに異常ある等）	アップロード対象ファイル、XMLファイルエラー
0003	リモートファイルエラー（対象が存在しない、ファイルに異常ある等）	ダウンロード対象ファイルエラー
0004	ファイル転送エラー	
0005	転送先ディレクトリ作成エラー	アップロード/ダウンロード用のWindows共有ディレクトリ配下へのサブディレクトリ作成失敗
0007	存在しないユーザが設定されているユーザ/グループが見つからない	
0008	存在しないジョブパターン ID が設定されているシナリオが見つからない	
0009	ライセンスエラー	APIライセンスが存在しない
0011	ジョブが見つからない	
0012	ini ファイルが見つからない、ini ファイル内に必要な値が無い	job.iniが見つからない 必須項目が指定されていない 指定した値の書式が誤っている
0013	指定されたジョブは処理中	ダウンロード可能なコンポーネントがない 終了ステータスになっているコンポーネントがない
0014	指定されたジョブ、ジョブセットは変換終了済み	
0024	ジョブパターン権限エラー	
0028	投入ファイル名禁則文字エラー	ファイル名がsystem.filename.invalid_patternにマッチする場合にエラー
0999	その他想定外のエラー	
3001	XML検証エラー	XML文書構造が不正
3002	リクエスト情報不備	必須項目が指定されていない 指定した値の書式が不正
3003	ファイル名禁則文字エラー	Windowsで使用できない文字を含む、または使用できない名前の場合にエラー
3004	文字コード変換エラー	
3005	サーバ接続エラー	リクエスト先が見つからない
3006	サーバ通信タイムアウト	
3007	サーバ通信エラー	
3008	ローカルディレクトリ作成エラー	
3009	XMLスキーマ読み込みエラー	XML検証に用いるXMLスキーマ定義ファイルが見つからない、定義内容が適切でない、その他IO例外

コード	説明	備考
3501	後続コマンド起動エラー	シェル(cmd.exe)を利用したコマンドの起動に失敗する 実行ファイルがパス上に存在しない
3502	後続コマンド実行中エラー	コマンドの実行中に例外が発生
3503	後続コマンド標準エラー出力	コマンド終了後、標準エラー出力への出力が存在している
4001	リクエスト不正	必須項目が設定されていない、設定値の書式不正 (最大長等含む)
4002	ファイルエラー (対象が存在しない、ファイルに異常ある等)	アップロード対象ファイル、XMLファイルエラー
4003	リモートファイルエラー (対象が存在しない、ファイルに異常ある等)	ダウンロード対象ファイルエラー
4004	ファイル転送エラー	
4005	転送先ディレクトリ作成エラー	アップロード/ダウンロード用のWindows共有ディレクトリ配下へのサブディレクトリ作成失敗
4006	ログイン名またはパスワードが正しくない	
4007	ログイン無効ユーザ	
4008	セッション無効	
4009	アップロード用Windows共有ディレクトリ未登録	ServerConfigurationへの設定不足 FileServerへの設定不足
4010	ダウンロード用Windows共有ディレクトリ未登録	ServerConfigurationへの設定不足 FileServerへの設定不足
4011	権限エラー(ユーザ/グループ対象権限)	[JobLotRegister] ユーザ・グループに対するオペレーション権限付与権限がない (Read, PermitDownload) [共通] UserAdminに所属するユーザでの実行が必要
4012	権限エラー(オペレーション対象権限)	[JobEntry] オペレーションにジョブを投入できない (Edit/Delete) ジョブの設定を参照できない (Read) [JobState] ジョブのステータスを参照できない (Read) [JobDownload] ジョブの出力ファイルをダウンロードできない (Download) ジョブのログファイルをダウンロードできない (Edit/Delete) ジョブの中間ファイルをダウンロードできない (Edit/Delete) [JobStop] ジョブを強制終了できない (Edit/Delete) [共通] JobAdminsグループに属していないユーザが削除されたジョブを指定している

コード	説明	備考
4013	権限エラー(シナリオ対象権限)	[JobEntry] ジョブ投入にシナリオを使用できない
4014	権限エラー(業務種別対象権限)	[JobLotRegister] 業務種別からオペレーションを作成する権限がない
4017	オペレーションが見つからない	
4018	業務種別が見つからない	
4019	ファイルサーバが見つからない	
4020	文字コード変換エラー	ダウンロード対象のファイルの絶対パス中にダウンロード用の Windows 共有フォルダの文字コードへ変換できない文字が含まれる
4021	許可されていないIPアドレスからのアクセス	[サーバ設定]のbatch.client.addressに定義されていないIPアドレスからのリクエストを受信した場合
4022	ハードリミット制約エラー(空き容量不足)	リソースフォルダが属するボリュームの空き容量が[サーバ設定]のsystem.resource.hard_limit.size以下、または共有フォルダが属するボリュームの空き容量が[サーバ設定]のsystem.shared_directory.hard_limit.size以下になった場合
4101	オペレーション作成が禁止されている業務種別(デフォルト)	
4102	オペレーション属性定義が見つからない	
4103	オペレーション属性定義の選択肢が見つからない	
4104	オペレーション属性定義が編集不可	
4109	オペレーション名が重複している	
4110	オペレーション識別名が重複している	
4111	オペレーション属性の制約違反	
4201	投入先オペレーションで使用できないシナリオ	
4202	コンポーネントが見つからない	
4203	コンポーネントパラメータが見つからない	
4204	コンポーネントパラメータの選択肢が見つからない	
4205	コンポーネントパラメータが編集不可	
4206	コンポーネントパラメータの制約違反	

コード	説明	備考
4207	投入ファイル形式エラー	シナリオが入力として受け付けるファイル形式と、投入ファイルの形式が異なる 入力ファイルが必要なシナリオに、投入ファイルが指定されていない 入力ファイルがないシナリオに、投入ファイルが指定されている
4208	投入ファイル形式エラー(ファイルサーバ)	ファイルサーバ上のZIPファイルが指定されている
4209	ジョブ登録後処理コールバックエラー	ジョブ登録後に設定されたコールバックURLへのリクエストに対して応答がない
4210	ファイルセットが見つからない	指定したファイルセットIDのファイルセットが見つからない
4211	ファイルセットID検証用コードが一致しない	指定した検証用コードとファイルセットの検証用コードが一致しない
4300	取得件数上限超過	JobState でジョブID省略時、ログインユーザの登録ジョブ数が ASFALIS TransServer で規定するジョブ取得上限数を超過している
4402	ダウンロード可能なファイルがない	終了ステータスであるが、ダウンロード可能なファイルがない
4403	ファイルが削除されている	終了ステータスであり、ダウンロード可能なファイルが定義されているが、ファイルの実体が削除されている
4410	ジョブダウンロードは成功したが出力ファイルが "DOWNDIR" 上に見つからない	"job.ini" 内の "DOWNDIR" の設定が誤っている可能性がある
4502	指定されたジョブは削除済み	JobAdminsグループに属するユーザで実行した 上記以外の場合、権限エラー(オペレーション対象権限)となる
4601	オペレーション属性が見つからない	指定したオペレーションIDと属性のキーの組み合わせが誤っている
4602	オペレーション属性がファイル型でない	
4999	サーバエラー	

## 3. APIのパラメータ

各APIのリクエストおよび、レスポンスのパラメータは、以下の通りです。

### 3.1. ログイン(/login/authenticate)

- リクエスト

階層	物理名	説明	型	必須
1	login_name	ログインに使用するユーザのログイン名を指定する。	string	○
1	password	ログインに使用するユーザのパスワードを指定する。	string	○

- レスポンス

階層	物理名	説明	型	必須
1	result_code	処理結果を表すコードを返却する。 (処理結果コード一覧を参照)	string	○
1	authenticity_token	処理結果コードが正常終了の場合のみ返却する。 他の更新用APIを利用する際に、本値を入力パラメータとして送信する。	string	*1

\*1: 処理結果コードが正常終了の場合に設定される

## 3.2. オペレーション登録(/batch/job\_lot\_register)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_lot_template	-	object	○
2	id	オペレーションの元とする業務種別のIDを指定する。	int	○
1	job_lot	-	object	○
2	name	オペレーションに設定する名前を指定する。 文字列を指定した場合は、ASFALIS TransServerで最も優先度が高く設定されている言語における名前として登録される。言語毎に名前を指定する場合は、キーにロケール名、値に名前を指定したオブジェクトとして記述する。	string / object	○
2	unique_alias	オペレーションに設定する識別名を指定する。 (他のオペレーションと重複する時はエラーになる) (省略時、ASFALIS TransServerが自動的に設定する) サーバ設定system.use_job_lots_unique_aliasesがTRUEのとき設定可能	string	-
1	job_lot_properties	省略時、オペレーション属性定義の情報が使用される。	array of object	-
2	-	-	object	-
3	key_name	オペレーション属性の元とするオペレーション属性定義のキー名を指定する。	string	○
3	value_type	"password"を指定した場合、このオペレーション属性に設定した値は、ASFALIS TransServerの操作ログに出力される際、「***」にマスクされる。	string	-
3	value	オペレーション属性値を指定する。 オペレーション属性定義に設定された値を使用する場合、本項目を省略する。空値を指定する場合、本項目を空要素にする。	string	*1

階層	ノード名	説明	型	必須
		<p>属性値として以下の型を使用可能</p> <ul style="list-style-type: none"> <li>• 数値</li> <li>• テキスト</li> <li>• パスワード</li> <li>• テキストエリア</li> <li>• 真偽値</li> <li>• サブウィンドウ</li> <li>• ラジオボタン: 選択肢に無い値を指定するとエラー</li> <li>• プルダウン: 選択肢に無い値を指定するとエラー</li> <li>• 日付: 指定日のUTC0時0分を、YYYY-MM-DD hh:mm:ss形式で指定する。 (例) JSTのマシンで2014年5月1日を指定する場合: 2014-05-01 09:00:00</li> <li>• 日時: UTCでYYYY-MM-DD hh:mm:ss形式で指定する。</li> <li>• ファイル: 割り当てるファイルの batch.upload.file_server に指定したファイルサーバ以下のパスを指定する。ファイルが見つからない場合はエラーとなる。ファイル登録後も元の共有ディレクトリ上のファイルは削除されない。 (例) \\machine\path を batch.upload.file_server に指定しているとき、file.txt を指定すると \\machine\path\file.txt を割り当てる。</li> </ul>		
1	job_lot_permissions	<p>オペレーションに対するEdit/Delete権限を付与するログインユーザを指定する。 サーバ設定の system.permissions.object_permissions.enabled がTRUEの場合、ここで指定する権限に加えて、ログインユーザ・JobAdminsにEdit/Delete権限が自動的に付与される。また、system.permissions.object_permissions.enabled がFALSEの場合、ここで指定する権限にかかわらずログインユーザ・JobAdminsおよびNamedUsersにEdit/Delete権限が付与される。</p>	array of object	-
2	-	-	object	-

階層	ノード名	説明	型	必須
3	permission_code	作成するオペレーションに対する権限の種別をコード指定する。 1: Read / 2: Download / 3: Edit/Delete	int	○
3	login_name	ユーザに対して権限を付与する場合、本項目に対象ユーザのログイン名を指定する。	array of string	-
3	group_id	グループに対して権限を付与する場合、本項目にグループのIDを指定する。	array of int	-
3	group_unique_aliases (group_internal_name)	グループに対して権限を付与する場合、本項目にグループの識別名 (unique_aliases) を指定する。	array of string	-

\*1: 省略時は空値を設定することと同等

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	job_lot	-	object	*1
2	id	登録されたオペレーションのIDを返却する。	int	○
2	unique_alias	登録されたオペレーションの識別名を返却する。 入力でオペレーション識別名を指定した場合は、指定した文字列がそのまま返却される。 入力でオペレーション識別名を指定しない場合は、ASFALIS TransServerが自動的に払い出した識別名が返却される。	string	○

\*1: 処理結果コードが正常終了の場合に設定される



- サンプル1

- リクエスト

ID=2 の業務種別にExampleOperation オペレーションを作成し、user1,user2 にダウンロード権限を付与する

```
{
  "authenticity_token": "*****"(認証トークン),
  "job_lot_template":
    {
      "id": 2
    },
  "job_lot":
    {
      "name": "ExampleOperation"
    },
  "job_lot_permissions":
    [
      {
        "login_name": ["user1", "user2"],
        "permission_code": 2
      }
    ]
}
```

- レスポンス

オペレーション作成、権限付与成功

```
{
  "job_lot":
    {
      "unique_alias": "joblot_2",
      "id": 3
    },
  "result_code": "0000"
}
```

- サンプル2

- リクエスト

ID=3 の業務種別にサンプル（日本語名） / Sample（英語名）オペレーションを作成する

```
{
  "authenticity_token": "*****"(認証トークン),
  "job_lot_template":
    {
      "id": 3
    },
  "job_lot":
    {
      "name": {
        "ja-JP": "サンプル",
        "en-US": "Sample"
      }
    }
}
```

- レスポンス

作成成功

```
{
  "job_lot":
    {
      "unique_alias": "joblot_1_20160810_110130",
      "id": 8
    },
  "result_code": "0000"
}
```

### 3.3. オペレーション更新(/batch/job\_lot\_update)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_lot	-	object	○
2	job_lot_id	更新するオペレーションのID	int	○
2	name	オペレーションに設定する名前を指定する。 文字列を指定した場合は、ASFALIS TransServerで最も優先度が高く設定されている言語における名前として登録される。言語毎に名前を指定する場合は、キーにロケール名、値に名前を指定したオブジェクトとして記述する。（省略時は更新されない）	string / object	-
2	unique_alias	オペレーションに設定する識別名を指定する。（他のオペレーションと重複する時はエラーになる）（省略時は更新されない） サーバ設定system.use_job_lots_unique_aliasesがTRUEのとき設定可能	string	-
2	is_erased	オペレーションの無効化、取戻を指定する。無効化する場合trueを指定する。	boolean	-
1	job_lot_properties	更新するオペレーションの属性を指定する。（省略時は属性が更新されない）	array of object	-
2	-	-	object	-
3	key_name	オペレーション属性の元とするオペレーション属性定義のキー名を指定する。	string	○
3	value_type	"password"を指定した場合、このオペレーション属性に設定した値は、ASFALIS TransServerの操作ログに出力される際、「**」にマスクされる。	string	-
3	value	オペレーション属性値を指定する。 オペレーション属性定義に設定された値を使用する場合、本項目を省略する。 空値を指定する場合、本項目を空要素にする。	string	*1

階層	ノード名	説明	型	必須
		<p>属性値として以下の型を使用可能</p> <ul style="list-style-type: none"> <li>• 数値</li> <li>• テキスト</li> <li>• パスワード</li> <li>• テキストエリア</li> <li>• 真偽値</li> <li>• サブウィンドウ</li> <li>• ラジオボタン: 選択肢に無い値を指定するとエラー</li> <li>• プルダウン: 選択肢に無い値を指定するとエラー</li> <li>• 日付: 指定日のUTC0時0分を、YYYY-MM-DD hh:mm:ss形式で指定する。 (例) JSTのマシンで2014年5月1日を指定する場合: 2014-05-01 09:00:00</li> <li>• 日時: UTCでYYYY-MM-DD hh:mm:ss形式で指定する。</li> <li>• ファイル: 割り当てるファイル のbatch.upload.file_serverに指定したファイルサーバ以下のパスを指定する。ファイルが見つからない場合はエラーとなる。ファイル登録後も元の共有ディレクトリ上のファイルは削除されない。 (例) \\machine\path をbatch.upload.file_serverに指定しているとき、file.txtを指定すると\\machine\path\file.txt を割り当てる。</li> </ul>		
1	job_lot_permissions	<p>オペレーションに対するEdit/Delete権限を付与するログインユーザを指定する。 サーバ設定のsystem.permissions.object_permissions.enabledがTRUEの場合、ここで指定する権限に加えて、ログインユーザ・JobAdminsにEdit/Delete権限が自動的に付与される。また、system.permissions.object_permissions.enabledがFALSEの場合、ここで指定する権限にかかわらずログインユーザ・JobAdminsおよびNamedUsersにEdit/Delete権限が付与される。</p>	array of object	-
2	-	-	object	-

階層	ノード名	説明	型	必須
3	permission_code	作成するオペレーションに対する権限の種別をコード指定する。 1: Read / 2: Download / 3: Edit/Delete	int	○
3	login_name	ユーザに対して権限を付与する場合、本項目に対象ユーザのログイン名を指定する。	array of string	-
3	group_id	グループに対して権限を付与する場合、本項目にグループのIDを指定する。	array of int	-
3	group_unique_aliases (group_internal_name)	グループに対して権限を付与する場合、本項目にグループの識別名 (unique_aliases) を指定する。	array of string	-

\*1: 省略時は空値を設定することと同等

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	job_lot	-	object	*1
2	id	登録されたオペレーションのIDを返却する。	int	○
2	unique_alias	登録されたオペレーションの識別名を返却する。 入力でオペレーション識別名を指定した場合は、指定した文字列がそのまま返却される。 入力でオペレーション識別名を指定しない場合は、ASFALIS TransServerが自動的に払い出した識別名が返却される。	string	○

\*1: 処理結果コードが正常終了の場合に設定される

- サンプル1

- リクエスト

ID=2 の業務種別にExampleOperation オペレーションを作成し、user1,user2 にダウンロード権限を付与する

```
{
  "authenticity_token": "*****"(認証トークン),
  "job_lot":
    {
      "job_lot_id": 1
    },
  "job_lot_properties":
    [
      {
        "key_name": "text_property",
        "value": "abc"
      },
      {
        "key_name": "file_property",
        "value": "file.txt"
      }
    ]
}
```

- レスポンス

オペレーション作成、権限付与成功

```
{
  "job_lot":
    {
      "unique_alias": "joblot_1",
      "id": 1
    },
  "result_code": "0000"
}
```

- サンプル2

- リクエスト

ID=8 のオペレーションの名前を 更新済み（日本語名） / Updated（英語名）に更新する

```
{
  "authenticity_token": "*****"(認証トークン),
  "job_lot":
    {
      "job_lot_id": 8
      "name": {
        "ja-JP": "更新済み",
        "en-US": " Updated "
      }
    }
}
```

- レスポンス

更新成功

```
{
  "job_lot":
    {
      "unique_alias": "joblot_1_20160810_110130",
      "id": 8
    },
  "result_code": "0000"
}
```

### 3.4. ジョブ登録(/batch/job\_entry)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	file_type	0: Part / 1: Assy	int	-
1	file_server_id	ASFALIS TransServerに登録されたファイルサーバのIDを指定する。 (FTPサーバまたはWindowsファイルサーバ)	int	*1a
1	file_server_dir	ファイルサーバIDで指定したファイルサーバの、入力ファイルが配置されているディレクトリを相対パスで指定する。	string	*1a
1	via_updir	ASFALIS TransServerに登録された入力ファイルアップロード用のWindowsファイルサーバを使用する場合、trueを指定する。	boolean	*1b
1	input_files	入力ファイル	array of object	*2a
2	-		object	○
3	input_file_path	ファイルサーバIDを指定した場合、ファイル名を指定する。 アップロード用Windows共有ディレクトリ使用フラグを指定した場合、当該ウィンドウズ共有ディレクトリからの相対パスで、入力ファイルを指定する。	string	○
3	top_assy	変換タイプが1:Assyのとき、トップアセンブリに指定するファイルに対してtrueを指定する。	boolean	-
1	input_file_sets	入力ファイルセット	array of object	*2b
2	-	-	object	○
3	file_set_id	投入するモデルのファイルセットIDを指定する。	int	○
3	embedded_code	投入するモデルのファイルセットID検証用コードを指定する。	string	○
1	input_file_sets_list	入力ファイルセットのリスト	array of array of object	*2c
2	-	-	array of object	○
3	-	-	object	○
4	file_set_id	投入するモデルのファイルセットIDを指定する。	int	○
4	embedded_code	投入するモデルのファイルセットID検証用コードを指定する。	string	○



階層	ノード名	説明	型	必須
4	component_instance_name	ファイルの投入先とするコンポーネントインスタンス名を指定する。 なお「コンポーネントインスタンス名」とは、シナリオ編集やジョブ詳細ページにて、薄い灰色の文字で確認できる識別名を指す。 例: parasolid2enf	string	○
4	input_key	投入するコンポーネントの入力パラメータのうち、IO TypeがinであるKeyを指定。 入力パラメータはコンポーネント詳細ページにて確認可能。	string	*3
1	job_lot_id	ジョブの投入先とするオペレーションのIDを指定する。	Int	○
1	src_job_id	既存のジョブのパラメータ設定を使用する場合、そのジョブのIDを指定する。	int	*4a
1	scenario_id	コンポーネントパラメータ設定を新規に行う場合、使用するシナリオのIDを指定する。	int	*4b
1	components	シナリオIDを指定して本項目を省略時、シナリオに設定された各パラメータのデフォルト値を使用する。	array of object	-
2	-	-	object	-
3	name	シナリオで選択したコンポーネントに対して設定されている一意な識別名を指定する。	string	○
3	parameters	-	array of object	-
4	-	-	object	-
5	key_name	コンポーネントのパラメータを一意に識別するキー名を指定する。	string	○
5	value_type	"password"を指定した場合、このパラメータに設定した値は、ASFALIS TransServerの操作ログに出力される際、「****」にマスクされる。	string	-
5	file_server_id	ASFALIS TransServerに登録されたファイルサーバのIDを指定する。 (FTPサーバまたはWindowsファイルサーバ)	int	
5	directory_path	ファイルサーバIDで指定したファイルサーバの、入力ファイルが配置されているディレクトリを相対パスで指定する。	string	

階層	ノード名	説明	型	必須
5	value	number: 直接指定 text: 直接指定 password: 直接指定 textarea: 直接指定 radiobutton: 選択肢の設定値を指定 boolean: 真の場合1、偽の場合0を指定 bitflag: 各フラグのON(1)とOFF(0)を連結して指定 file: Windows共有ディレクトリ上に配置されたファイルのUNCパスを指定 user: ログイン名を指定 combobox: 選択肢の設定値または直接指定 pulldown: 選択肢の設定値を指定 ftp_param: ファイルサーバID:相対パス cifs_param: ファイルサーバID:相対パス (例) 1:test  複数値パラメータの場合、本項目は設定しない。	string	-
5	multiple_values	-	array of string (object)	-
6	-	number: 直接指定 text: 直接指定 user: ログイン名を指定 file: Windows共有ディレクトリ上に配置されたファイルのUNCパスを指定 file_with_key: 後述のパラメータを持つオブジェクトを指定	string (object)	-
7	key_name	(file_with_keyのみ) キー名を指定する。	string	○
7	file_path	(file_with_keyのみ) Windows共有ディレクトリ上に配置されたファイルのUNCパスを指定する。	string	○
3	sub_components	本項目を省略時、シナリオに設定された各パラメータのデフォルト値を使用する。	array of object	-
4	-	-	object	-
5	name	シナリオで選択したサブコンポーネントに対して設定されている一意な識別名を指定する。	string	○
5	parameters	-	array of object	-
6	-	-	object	-
7	key_name	サブコンポーネントのパラメータを一意に識別するキー名を指定する。	string	○
7	file_server_id	ASFALIS TransServerに登録されたファイルサーバのIDを指定する。(FTPサーバまたはWindowsファイルサーバ)	int	
7	directory_path	ファイルサーバIDで指定したファイルサーバの、入力ファイルが配置されているディレクトリを相対パスで指定する。	string	

階層	ノード名	説明	型	必須
7	value	number: 直接指定 text: 直接指定 password: 直接指定 textarea: 直接指定 radiobutton: 選択肢の設定値を指定 boolean: 真の場合1、偽の場合0を指定 bitflag: 各フラグのON(1)とOFF(0)を連結して指定 file: Windows共有ディレクトリ上に配置されたファイルのUNCパスを指定 user: ログイン名を指定 combobox: 選択肢の設定値または直接指定 pulldown: 選択肢の設定値を指定 ftp_param: ファイルサーバID:相対パス cifs_param: ファイルサーバID:相対パス  複数値パラメータの場合、本項目は設定しない。	string	-
7	multiple_values	-	array of object	-
8	-	-	object	-
9	value	number: 直接指定 text: 直接指定 user: ログイン名を指定 file: Windows共有ディレクトリ上に配置されたファイルのUNCパスを指定	string	○
1	job_option	省略時、各項目の初期値で登録される。	object	-
2	priority	1: 低 / 2: 中 / 3: 高 / 4: 最高 省略時: 1: 低	int	-
2	is_nightly	ジョブの実行を夜間に限定する場合に指定する。 1: 夜間のみ実行 / 0: 常時実行 省略時: 0: 常時実行	boolean	-
2	description	省略時: 空文字	string	-

\*1: シナリオに入力ファイルが無い場合を除き1aの全て、または1bのどちらかが必須

\*2: シナリオに入力ファイルが無い場合を除き2a, 2b, 2cのいずれかが必須

\*3: 投入するコンポーネントの入力ファイルが2つ以上ある場合のみ必須

\*4: 4a すべてまたは 4b すべてのどちらかが必須

#### • レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	jobs	-	array of object	*1
2	-	-	object	○
3	id	登録されたジョブのジョブID。	int	○

階層	ノード名	説明	型	必須
3	unique_alias	ASFALIS TransServerが自動採番したジョブのジョブ識別名。 形式: "job_\${ジョブID}_\${ジョブ登録日時:YYYYMMDD_hhmmss}" (既に使用されている場合、 "job_\${ジョブID}_\${ジョブ登録日時:YYYYMMDD_hhmmss}_\${シーケンス番号}")	string	○

\*1: 処理結果コードが正常終了の場合に設定される

- サンプル1

- リクエスト

ID=6 の外部ファイルサーバのJobEntry ディレクトリにあるasm1.igsファイルを入力として、ID=2 のシナリオを用いてID=3 のオペレーションにジョブを投入する

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "file_server_id": 6,
  "file_server_dir": "JobEntry",
  "input_files":
  [
    {
      "input_file_path": "asm1.igs",
    }
  ],
  "job_lot_id": 3,
  "scenario_id": 2
}
```

- レスポンス

投入成功

```
{
  "result_code": "0000",
  "jobs":
  [
    {
      "unique_alias": "job_3_20121116_111320",
      "id": 3
    }
  ]
}
```

- サンプル.2

- リクエスト

ENF 編集のパラメータファイルリストにパラメータを指定してジョブを投入する。  
外部ファイルサーバ上の"sample.xml"ファイルを"extract\_option"というキー名、および"sample.txt"ファイルを"sample"というキー名で指定する。

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "file_type": 0,
  "via_updir": true,
  "input_files": [
    {
      "input_file_path": "sample.igs"
    }
  ],
  "components": [
    {
      "key_name": "parameterfile_list",
      "value_type": "file_with_key",
      "multiple_values": [
        {
          "key_name": "extract_option",
          "file_path": "sample.xml"
        },
        {
          "key_name": "sample",
          "file_path": "sample.txt"
        }
      ]
    }
  ],
  "job_lot_id": 2,
  "scenario_id": 4
}
```

- レスポンス

投入成功

```
{
  "result_code": "0000",
  "jobs":
  [
    {
      "id": 6,
      "unique_alias": "job_6_20161202_162107"
    },
    {
      "id": 7,
      "unique_alias": "job_7_20161202_162108"
    }
  ]
}
```

- サンプル.3

- リクエスト

オペレーションID=2,シナリオID=1のシナリオに対して、igesインポートの入力をID=10のファイルセット、Creo Parametricインポートの入力をID=11のファイルセットとするジョブ、およびigesインポートの入力をID=20のファイルセット、Creo Parametricインポートの入力をID=21のファイルセットとするジョブの2つを投入する

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "scenario_id": 1,
  "job_lot_id": 2,
  "input_file_sets_list":
  [
    [
      {
        "file_set_id": 10,
        "embedded_code": "*****" (ファイルセットID検証用コード),
        "component_instance_name": "iges2enf",
        "input_key": "inputfile"
      },
      {
        "file_set_id": 11,
        "embedded_code": "*****" (ファイルセットID検証用コード),
        "component_instance_name": "proe2enf",
        "input_key": "inputfile"
      }
    ],
    [
      {
        "file_set_id": 20,
        "embedded_code": "*****" (ファイルセットID検証用コード),
        "component_instance_name": "iges2enf",
        "input_key": "inputfile"
      },
      {
        "file_set_id": 21,
        "embedded_code": "*****" (ファイルセットID検証用コード),
        "component_instance_name": "proe2enf",
        "input_key": "inputfile"
      }
    ]
  ]
}
```



- レスポンス

投入成功

```
{
  "result_code": "0000",
  "jobs":
  [
    {
      "id": 4,
      "unique_alias": "job_91_20140108_162107"
    },
    {
      "id": 5,
      "unique_alias": "job_92_20140108_162108"
    }
  ]
}
```

## 3.5. ジョブステータス取得(/batch/job\_state)

### • リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_id	ステータスを取得する対象のジョブIDを指定する。省略時、ログインユーザが登録したジョブが対象となる。	int	-
1	job_id_list	ステータスを取得する対象のジョブIDを配列で指定する。省略時、ログインユーザが登録したジョブが対象となる。	array of int	-

### • レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	jobs	-	array of object	*1
2	-	-	(object)	-
3	id	ジョブのID。	int	○
3	created_at	形式: YYYY-MM-DD hh:mm:ssZ	string	○
3	input_filename	入力ファイルを受け取らないシナリオから作成されたジョブの場合、空文字が設定される。	string	○
3	status	ジョブのステータス。 1: 未処理 / 2: 処理中 / 3: 正常終了 / 4: 異常終了 / 5: 強制終了	int	○

\*1: 処理結果コードが正常終了の場合に設定される

(例) 「[ログイン、ログアウト以外のAPIの利用](#)」の例をご参照ください。

## 3.6. ジョブダウンロード(/batch/job\_download)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_id	ダウンロード対象のジョブのIDを指定する。	int	○
1	download_dir	ASFALIS TransServerがダウンロード対象ファイルを配置するパスを、バッチとASFALIS TransServerで共有するWindows共有ディレクトリ（サーバ設定のbatch.download.file_serverに指定した外部ファイルサーバのパス）からの相対パスで指定する。	string	○
1	job_output_flg	ジョブの出力ファイルとして指定されている変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	log_flg	ログファイルをダウンロードする場合にtrueを指定する。	boolean	*1
1	component_output_flg	ジョブの出力ファイルとして指定されていない変換結果ファイルをダウンロードする場合にtrueを指定する。	boolean	*1
1	child_job_output_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブの出力ファイルとして指定されている変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	child_job_log_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブのログファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	child_job_component_output_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブの出力ファイルとして指定されていない変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1

\*1: どれか1つは必須

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	top_file_info_list	出力ファイルのトップアセンブリの情報	array of object	○
2	-	-	object	-
3	path	トップアセンブリファイルが存在するパス。 download_dirで指定したディレクトリからの 相対パス。	string	○
3	top_file	トップアセンブリファイルのファイル名	string	○
1	zip_flg	(カスタマイズ時のみ) ダウンロードカスタマイズサーバから、ダウンロード形式にzipが指定された場合にtrueを返却する。	boolean	-
1	zip_file_name	(カスタマイズ時のみ) ダウンロードカスタマイズサーバから指定されたzipファイル名を返却する。	string	-

- サンプル1

- リクエスト

ID=3のジョブの出力ファイルを、JobDownloadディレクトリにダウンロードする

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "job_id": 3,
  "download_dir": "JobDownload",
  "job_output_flg": true
}
```

- レスポンス

ダウンロード成功

```
{
  "result_code": "0000",
  "zip_flg": false,
  "zip_file_name": null
}
```

- サンプル2

- リクエスト

ID=109のジョブの出力ファイルを、JobDownloadディレクトリにダウンロードする

```
{
  "authenticity_token": "*****"（認証トークン）,
  "job_id": 109,
  "download_dir": "JobDownload",
  "job_output_flg": true
}
```

- レスポンス

ダウンロード成功

```
{
  "zip_flg": false,
  "zip_file_name": nil,
  "result_code": "0000",
  "top_file_info_list":
  [
    {
      "path" "J109_sample/C2_enf2nx/OUTPUT",
      "top_file" "sample.prt"
    }
  ]
}
```

## 3.7. ジョブダウンロード(直接)

### (/batch/job\_download\_direct)

- リクエスト HTTP GETメソッドで送信する

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_id	ダウンロード対象のジョブのIDを指定する。	int	○
1	job_output_flg	ジョブの出力ファイルとして指定されている変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	log_flg	ログファイルをダウンロードする場合にtrueを指定する。	boolean	*1
1	component_output_flg	ジョブの出力ファイルとして指定されていない変換結果ファイルをダウンロードする場合にtrueを指定する。	boolean	*1
1	child_job_output_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブの出力ファイルとして指定されている変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	child_job_log_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブのログファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1
1	child_job_component_output_flg	job_idで指定したジョブに子ジョブが存在する時に有効。 子ジョブの出力ファイルとして指定されていない変換結果ファイルをダウンロード対象に含める場合にtrueを指定する。	boolean	*1

\*1: どれか1つは必須

- レスポンス(成功) HTTPステータスコード 200 を返却する
  - 結果が単数ファイルの場合: ファイルそのものを返却する
  - 結果が複数ファイルの場合: 該当するファイルをアーカイブしたzipファイルを返却する
  - ダウンロードファイルの名前は、UTF-8としてパーセントエンコーディングされてレスポンスヘッダ中のContent-Disposition に含めて送られます。(RFC6266 準拠の形式)
- レスポンス(失敗) HTTPステータスコード 404 を返却する

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○

- サンプル1

## 。 リクエスト

ID=3のジョブの出力ファイルのみをダウンロードする

```
GET http:// (サーバ名):(Port番号)/batch/job_download_direct HTTP/1.1
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Host: (サーバ名):(Port番号)
Content-Type: application/json; charset=utf-8
Connection: Keep-Alive

{
  "authenticity_token": "*****" (認証トークン) ,
  "job_id": 3,
  "job_output_flg": true
}
```

## 。 レスポンス

ダウンロード成功

```
HTTP/1.1 200 OK
Date: Wed, 19 Oct 2016 09:00:00 GMT
Server: nginx/1.2.9
Content-Type: text/plain; charset=utf-8
Content-Length: 1024
Content-Disposition: attachment;
filename="sample.stp"; filename*=utf-8'sample.stp

(File Data: 1024 bytes)
```

- サンプル2

- リクエスト

ID=3のジョブの出力ファイル・ログファイル・コンポーネントファイルをダウンロードする

```
GET http:// (サーバ名):(Port番号)/batch/job_download_direct HTTP/1.1
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Host: (サーバ名):(Port番号)
Content-Type: application/json; charset=utf-8
Connection: Keep-Alive

{
  "authenticity_token": "*****" (認証トークン) ,
  "job_id": 3,
  "job_output_flg": true,
  "log_flg": true,
  "component_output_flg": true
}
```

- レスポンス

ダウンロード成功

```
HTTP/1.1 200 OK
Date: Wed, 19 Oct 2016 09:00:00 GMT
Server: nginx/1.2.9
Content-Type: application/zip
Content-Length: 4096
Content-Disposition: attachment;
filename="J3_sample.zip";
filename*=utf-8'J3_sample.zip

(File Data: 4096 bytes)
```



- サンプル3

- リクエスト

ID=4のジョブの出力ファイルをダウンロードする

```
GET http:// (サーバ名):(Port番号)/batch/job_download_direct HTTP/1.1
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Host: (サーバ名):(Port番号)
Content-Type: application/json; charset=utf-8
Connection: Keep-Alive

{
  "authenticity_token": "*****" (認証トークン) ,
  "job_id": 4,
  "job_output_flg": true
}
```

- レスポンス

ファイルが削除されていてダウンロードできなかった

```
HTTP/1.1 404 NOT FOUND
Date: Wed, 19 Oct 2016 09:00:00 GMT
Server: nginx/1.2.9
Content-Type: application/json; charset=utf-8

{
  "result_code": "4403"
}
```

## 3.8. ジョブ中断(/batch/job\_stop)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_id	中断（強制終了）する対象のジョブIDを指定する。	int	○

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○

- サンプル

- リクエスト

ID=4のジョブを強制終了する

```
{
  "authenticity_token": "*****"（認証トークン）,
  "job_id": 4
}
```

※ ※

### 3.9. ファイルアップロード(/batch/upload)

- ファイルアップロード(/batch/upload)に限り、アクセストークンを利用した認証はできません。セッション情報と認証トークンを使用してください。
- リクエスト  
multipart/form-data 形式のリクエストをPOSTで送信してファイルアップロードを実施します。アップロードするファイルの内容に応じて、ヘッダとボディを次のように設定してリクエストを送信してください。

- ヘッダ

Content-Type に含まれるboundaryは送信するデータ内の区切りを表す文字列です。同一リクエスト内では同じ文字列を指定してください。また、送信するデータに含まれないような文字列を指定してください。

X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID には、同じファイルを扱うリクエスト間で共通の文字列を指定してください。

分割アップロード時は、送信中のデータの範囲情報を渡すため、ヘッダにContent-Rangeを指定してください。

- 一括アップロード

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID:
*** (1ファイル毎にユニークな文字列)
Content-Disposition: attachment; filename="*** (ファイル名)"
Content-Type: multipart/form-data; boundary=***
X-CSRF-Token: *** (認証トークン)
Cookie: _K2Dashboard_session=*** (セッション情報)
```

- 分割アップロード

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID:
*** (1ファイル毎にユニークな文字列)
Content-Disposition: attachment; filename="*** (ファイル名)"
Content-Type: multipart/form-data; boundary=***
Content-Range: bytes ***-***/***(バイトごとに区切って何番地から-
何番地までのデータ/全体のバイト数 *1)
X-CSRF-Token: *** (認証トークン)
Cookie: _K2Dashboard_session=*** (セッション情報)
```

\*1: 例えば2050Byteのファイルを1000Byteずつ送信する場合、リクエストは3回に分かれます。それぞれのリクエストのContent-Rangeは 0-999/2050, 1000-1999/2050, 2000-2049/2050 のように指定してください。

- ボディ

ファイルアップロードはパラメータ名 upload でのみ受け付けています。ボディの Content-

Disposition 内で name="upload" を指定してください。

```
--*** (boundaryの文字列)
Content-Disposition: form-data; name="upload"; filename="*** (ファイル名)"
Content-Type: application/octet-stream
(改行)
*** (ファイルの中身)
--*** (boundaryの文字列)--
```

- レスポンス

- 分割アップロードの継続中

```
{
  "result_code": "0000",
  "status": "continue_partial_upload",
  "message": ""
}
```

- 一括アップロード/分割アップロードの完了時

- 単一ファイルの場合

upload\_files 以下のノードのidはアップロードが完了したファイルの一時IDです。

```
{
  "result_code": "0000",
  "status": "success",
  "message": "",
  "upload_files": [
    {
      "id": "4349186a-77a9-4113-9c32-30d74799c684",
      "size": 2050,
      "filename": "Sample.igs",
      "state": "success"
    }
  ]
}
```

- zipファイルの場合

- upload\_files 以下のノードのidはアップロードが完了したファイルの一時IDです。
- entries 以下のノードはアップロードしたzipに含まれていたファイルであり、derivation\_id で紐づいています。zipの一時IDのみでzip内のファイルをまとめてファイルセット登録することが可能です。

```
{
  "result_code": "0000",
  "status": "success",
  "message": "",
  "upload_files": [
    {
      "id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
      "size": 36276,
      "filename": "Assembly.zip",
      "state": "success",
      "entries": [
        {
          "id": "fae7be51-8915-49a2-8def-9b435862ab98",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": 54100,
          "filename": "part1.CATPart"
        },
        {
          "id": "d564303a-5a87-43ce-b206-caf13d57b2de",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": 56040,
          "filename": "part2.CATPart"
        },
        {
          "id": "eb78d8f7-81ab-41aa-ac89-aa7f24bbeb82",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": 16505,
          "filename": "Assembly.CATProduct"
        }
      ]
    }
  ]
}
```

- サンプル1

- リクエスト

ファイルを1000byteごとに分割アップロードする。

- ヘッダ

```
// 1回目のリクエスト
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID: bf437123-b190-4cce-9785-ab128ac3c6f2
Content-Disposition: attachment; filename="Cone.igs"
Content-Type: multipart/form-data; boundary=Rl62qoch
Content-Range: bytes 0-999/2050
X-CSRF-Token: ***(認証トークン)
Cookie: _K2Dashboard_session=***(セッション情報)

// 2回目のリクエスト
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID: bf437123-b190-4cce-9785-ab128ac3c6f2
Content-Disposition: attachment; filename="Cone.igs"
Content-Type: multipart/form-data; boundary=FgPcMaDD
Content-Range: bytes 1000-1999/2050
X-CSRF-Token: ***(認証トークン)
Cookie: _K2Dashboard_session=***(セッション情報)

// 3回目のリクエスト
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID: bf437123-b190-4cce-9785-ab128ac3c6f2
Content-Disposition: attachment; filename="Cone.igs"
Content-Type: multipart/form-data; boundary=0hwiogqW
Content-Range: bytes 2000-2049/2050
X-CSRF-Token: ***(認証トークン)
Cookie: _K2Dashboard_session=***(セッション情報)
```

- ボディ

```
// 1回目のリクエスト
--Rl62qocH
Content-Disposition: form-data; name="upload"; filename="Cone.igs"
Content-Type: application/octet-stream

***(ファイルの中身)
--Rl62qocH--

// 2回目のリクエスト
--FgPcMaDD
Content-Disposition: form-data; name="upload"; filename="Cone.igs"
Content-Type: application/octet-stream

***(ファイルの中身)
--FgPcMaDD--

// 3回目のリクエスト
--0hwiogqW
Content-Disposition: form-data; name="upload"; filename="Cone.igs"
Content-Type: application/octet-stream

***(ファイルの中身)
--0hwiogqW--
```

### 。レスポンス

ファイルのアップロードを3回に分けて実行後、アップロードが完了し、ファイルは一時ID="4349186a-77a9-4113-9c32-30d74799c684"のファイルとして利用可能になっている状態。

```
// 1回目のレスポンス
{
  "result_code": "0000",
  "status": "continue_partial_upload",
  "message": ""
}

// 2回目のレスポンス
{
  "result_code": "0000",
  "status": "continue_partial_upload",
  "message": ""
}

// 3回目のレスポンス
{
  "result_code": "0000",
  "status": "success",
  "message": "",
  "upload_files": [
    {
      "id": "4349186a-77a9-4113-9c32-30d74799c684",
      "size": 2050,
      "filename": "Sample.igs",
      "state": "success"
    }
  ]
}
```



- サンプル2

- リクエスト

zipファイルを一括アップロードする。

- ヘッダ

```
X-ASFALIS-TRANSSERVER-BATCH: TRUE
X-ASFALIS-TRANSSERVER-UPLOAD-REQUEST-ID: 2d09f4ea-3f3a-4d4d-ab35-
5f6bb590d728
Content-Disposition: attachment; filename="wheelset.zip"
Content-Type: multipart/form-data; boundary=Yv0C7zjh
X-CSRF-Token: *** (認証トークン)
Cookie: _K2Dashboard_session=*** (セッション情報)
```

- ボディ

```
--Yv0C7zjh
Content-Disposition: form-data; name="upload"; filename="wheelset.zip"
Content-Type: application/octet-stream

*** (ファイルの中身)
--Yv0C7zjh--
```

## 。レスポンス

zipファイルのアップロードが完了し、zipファイルは一時ID="86c0fa0e-5409-40b9-9521-9e7a6a41abd3"のファイルとして利用可能になっている状態。

```
{
  "result_code": "0000",
  "status": "success",
  "message": "",
  "upload_files": [
    {
      "id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
      "size": 36276,
      "filename": "wheelset.zip",
      "state": "success",
      "entries": [
        {
          "id": "fae7be51-8915-49a2-8def-9b435862ab98",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": "54100",
          "filename": "bar.CATPart"
        },
        {
          "id": "d564303a-5a87-43ce-b206-caf13d57b2de",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": "56040",
          "filename": "wheel.CATPart"
        },
        {
          "id": "eb78d8f7-81ab-41aa-ac89-aa7f24bbeb82",
          "derivation_id": "86c0fa0e-5409-40b9-9521-9e7a6a41abd3",
          "size": "16505",
          "filename": "wheelset.CATProduct"
        }
      ]
    }
  ]
}
```

## 3.10. ファイルセット登録(/batch/file\_set\_register)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	mime_type	作成するファイルセットのMIMEタイプを指定する。	string	○
1	Input_file_sets	入力ファイルセット	array of object	○
2	-	-	object	○
3	temporary_file_set_id	APIのリクエスト・レスポンス間で一意のIDを指定する。レスポンスの際にファイルセットを指定するのに使用される。	string	○
3	file_type	0: Part / 1: Assy	int	-
3	file_server_id	ASFALIS TransServer に登録されたファイルサーバのIDを指定する。 (FTPサーバまたはWindowsファイルサーバ)	int	*1a
3	file_server_dir	ファイルサーバIDで指定したファイルサーバの、入力ファイルが配置されているディレクトリを相対パスで指定する。	string	*1a
3	via_updir	ASFALIS TransServer に登録された入力ファイルアップロード用のWindowsファイルサーバを使用する場合、trueを指定する。	boolean	*1b
3	input_files	入力ファイル	array of object	○
4	-	-	object	○
5	input_file_path	ファイルサーバIDを指定した場合:ファイル名を指定する。 アップロード用Windows共有ディレクトリ使用フラグを指定した場合:当該ウィンドウズ共有ディレクトリからの相対パスで、入力ファイルを指定する。	string	*2a
5	temporary_id	/batch/upload でアップロードしたファイルの一時IDを指定する。ファイルサーバの指定や登録がなくても利用が可能。	string	*2b
5	top_assy	トップアセンブリに指定するファイルに対してtrueを指定する。	boolean	-

\*1: input\_file\_path によるファイルの指定を含む場合、1a の全て、または 1b のどちらかが必須

\*2: 2a または 2b のどちらかが必須

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	file_sets	-	array of object	*1
2	-	-	object	○
3	temporary_file_set_id	テンポラリファイルセットID (リクエストで指定されたID)	string	○
3	id	登録されたファイルセットのID	int	○
3	embedded_code	ファイルセットID検証用コード。ファイルセットIDの改ざん防止のための文字列でジョブ投入時ファイルセットを入力する時に必要。	string	○

\*1: 処理結果コードが正常終了の場合に設定される

- サンプル

- リクエスト

ID=1のファイルサーバに登録されているデータ3つをアセンブリの1つのファイルセットとして登録する

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "mime_type": "application/nx",
  "input_file_sets":
  [
    {
      "temporary_file_set_id": "1",
      "file_server_id": 1,
      "file_server_dir": "NX/wheel_set",
      "file_type": 1,
      "input_files":
      [
        {
          "input_file_path": "wheel_set.prt",
          "top_assy": true
        },
        {
          "input_file_path": "wheel.prt"
        },
        {
          "input_file_path": "bar.prt"
        }
      ]
    }
  ]
}
```

## • レスポンス

ID=5のファイルセットとして登録成功

```
{
  "file_sets":
  [
    {
      "temporary_file_set_id": "1",
      "id": 5,
      "embedded_code": "*****_***_*****_*****_*****"
      (検証用コード)
    }
  ]
  "result_code": "0000"
}
```

## ジョブ投入時パラメータ指定ファイルによるパラメータ指定

ファイルセット登録時に拡張子が .atsparam のファイルを一緒に登録することで、そのファイルセットでジョブを投入するときの変換パラメータを上書きすることができます。このパラメータを指定するファイルのことを**ジョブ投入時パラメータ指定ファイル**と呼びます。

Creo Parametricの簡略表示名など、パラメータの値がモデルに依存するような場合に便利です。

- ジョブ投入時パラメータ指定ファイルの書式はJSON形式で以下のようになります。

階層	ノード名	説明	型	必須
1	format_version	1 を指定します	int	○
1	component_instance_list	-	array of object	○
2	-	-	object	-
3	name	コンポーネントインスタンス名-	string	○
3	parameter	APIのリクエスト・レスポンス間で一意のIDを指定する。レスポンスの際にファイルセットを指定するのに使用される。	object	○
4	パラメータ名(*1)	パラメータの値	(*2)	

\*1:各コンポーネントのパラメータ名を指定します。指定が無いパラメータは上書きされません。またユーザが編集可能なパラメータのみ更新されます。

\*2:パラメータごとに型が異なります。適切な型で指定してください。

- サンプル

- リクエスト

ID=1のファイルサーバに登録されているデータ3つをアセンブリの1つのファイルセットとして登録する

```
{
  "format_version": 1,
  "component_instance_list":
    [
      {
        "name": "iges2enf",
        "parameter":
          {
            "Stitch": 1,
            "StitchTol": 0.01
          }
      },
      {
        "name": "dek",
        "parameter":
          {
            "HealPartwise": 1
          }
      }
    ]
}
```

## 3.11. ファイル形式情報取得(/batch/mime\_type\_info)

### • リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	scenario_id	入力ファイルのMIMEタイプを取得する対象のシナリオを、シナリオIDで指定する。	int	*1
1	job_id	入力ファイルのMIMEタイプを取得する対象のシナリオを、ジョブIDで指定する。	int	*1

\*1: どちらか必須

### • レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	accept_input	シナリオが入力ファイルを受け取る場合にtrueを返却する。	boolean	*1
1	mime_type_list	シナリオの入力ファイルのMIMEタイプ名称を返却する。 例: application/enf	array of string	*2
1	mime_type	mime_type_list の先頭を返却する。 *a	string	*2
1	extensions_list	このMIMEタイプのファイルの拡張子パターンを、拡張正規表現パターンとして返却する。 例: ["enf", "enf_[a-zA-Z0-9]+", "drx_[a-zA-Z0-9]+"]	array of array of string	*2
1	extensions	extensions_list の先頭を返却する。 *a	array of string	*2
1	objective_flg_list	シナリオの入力ファイルの形式が非オブジェクト型の場合trueを、オブジェクト型の場合falseを返却する。	array of boolean	*2
1	objective_flg	objective_flg_list の先頭を返却する。 *a	boolean	*2
1	component_instance_name_list	シナリオの入力ファイルを受け取るコンポーネントインスタンス名を返却する。	array of string	*3
1	input_key_list	シナリオの入力ファイルのキーを返却する。	array of string	*3

\*a: EX5.2.5 との後方互換用。

\*1: 処理結果コードが正常終了の場合に設定される。

\*2: 入力ファイル有無フラグ(accept\_input)がtrueの場合に設定される。また、シナリオの入力ファイルが単一の場合は上の型、複数の場合は下の型となる。

\*3: シナリオの入力ファイルが複数の場合のみ設定される。job\_entry にて input\_file\_sets\_list を用いる場合、これらの値をパラメータとして使用。

- サンプル

- リクエスト

ID=2のシナリオの入力ファイルのMIMEタイプを取得する

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "scenario_id": 2
}
```

- レスポンス

取得成功

```
{
  "result_code": "0000",
  "accept_input": true,
  "mime_type_list": ["application/iges"],
  "extensions_list": [
    ["iges", "igs"]
  ],
  "objective_flg_list": [false],
  "mime_type": "application/iges",
  "extensions": ["iges", "igs"],
  "objective_flg": false,
  "component_instance_name_list": ["iges2enf"],
  "input_key_list": ["inputfile"]
}
```



## 3.12. ファイル型オペレーション属性ダウンロード (/batch/job\_lot\_attributes\_download)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_lot_id	属性を取得するオペレーションのID	int	○
1	download_dir	ASFALIS TransServerがダウンロード対象ファイルを配置するパスを、バッチとASFALIS TransServerで共有するWindows共有ディレクトリ（サーバ設定のbatch.download.file_serverに指定した外部ファイルサーバのパス）からの相対パスで指定する。	string	○
1	key_name_list	取得するファイル型のオペレーション属性のキー名を配列で指定する。 指定したキーが存在しない場合およびファイル型でない場合はエラーとなる。	array of string	○

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照	string	○

- サンプル

- リクエスト

ID=2のオペレーションのキー名file\_type\_attr\_1,file\_type\_attr\_2のファイル型属性をJobDownloadディレクトリにダウンロードする

```
{
  "authenticity_token": "*****"（認証トークン）,
  "job_lot_id": 2,
  "download_dir": "JobDownload",
  "key_name_list": ["file_type_attr_1", "file_type_attr_2"]
}
```

- レスポンス

ダウンロード成功

```
{
  "result_code": "0000"
}
```

### 3.13. ファイル型オペレーション属性ダウンロード(直接) (/batch/job\_lot\_attributes\_download\_direct)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	job_lot_id	属性を取得するオペレーションのIDを指定する。	int	○
1	key_name_list	取得するファイル型のオペレーション属性のキー名を配列で指定する。 指定したキーが存在しない場合およびファイル型でない場合はエラーとなる。	array of string	○

- レスポンス(成功)

HTTPステータスコード 200 を返却する

- 結果が単数ファイルの場合: ファイルそのものを返却する
- 結果が複数ファイルの場合: 該当するファイルをアーカイブしたzipファイルを返却する
- ダウンロードファイルの名前は UTF-8 としてパーセントエンコーディングされ、レスポンスヘッダ中の Content-Disposition に含めて送られる (RFC6266 準拠の形式)

- レスポンス(失敗)

HTTPステータスコード 404 を返却する

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照	string	○

- サンプル

- リクエスト

ID=2 のオペレーションのキー名 file\_type\_attr\_1 のファイル型属性をダウンロードする

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "job_lot_id": 2,
  "key_name_list": ["file_type_attr_1"]
}
```

```
GET http://(サーバ名):(Port番号)/batch/job_lot_attributes_download_direct
HTTP/1.1
X-ASFALIS-TRANSSERVER-BATCH: TRUE
Cookie: _K2Dashboard_session=***** (セッション情報)
Host: (サーバ名):(Port番号)
Content-Type: application/json; charset=utf-8
Connection: Keep-Alive
```

。レスポンス

ダウンロード成功

```
HTTP/1.1 200 OK
Date: (日時) GMT
Server: nginx/1.12.2
Content-Type: text/html
Content-Length: 2048
Content-Disposition: attachment; filename="sample.data"; filename*=utf-8'sample.data

(File Data: 2048 bytes)
```

## 3.14. ユーザ情報取得(/batch/users)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	user_id_list	情報を取得する対象のユーザをIDの配列で指定する。	array of int	*1
1	user_login_name_list	情報を取得する対象のユーザをログイン名の配列で指定する。	array of string	*1

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	users	配下のパラメータについては、管理者マニュアルのユーザインポートのパラメータ一覧を参照。（パスワードは出力時に保護されるため、参照先と形式が異なります。）	array of object	*1

\*1: 処理結果コードが正常終了の場合に設定される。

- サンプル

- リクエスト

ログイン名が sample\_user1 のユーザの情報を取得する。

```
{
  "authenticity_token": "*****"（認証トークン）,
  "user_login_name_list": ["sample_user1"]
}
```

- レスポンス  
取得成功

```
{
  "result_code": "0000",
  "users": [
    {
      "login_name": "sample_user1",
      "email": "sample_user1@email.com",
      "is_login_enable": true,
      "is_editable_email": true,
      "is_error_mail_recipient": true,
      "is_mail_component_mail_recipient": true,
      "is_all_parameter_open": true,
      "time_zone": "Tokyo",
      "is_enable_ldap": true,
      "ldap_account_name": "sample_user1",
      "external_owner": null,
      "encrypted_password": {
        "password_hash": "d680c8621a8e3d260c37c6c10b4cafe4",
        "password_salt": "lNQ48MgU"
      },
      "name": {
        "en-US": {
          "first_name": "Sample",
          "family_name": "User1",
          "name_format_type": "first_space_family"
        }
      },
      "memo": {
        "en-US": "sample_user1 memo"
      },
      "language_priority": [
        "en-US",
        "ja-JP",
        "de-DE"
      ],
      "character_code": "Shift_JIS"
    }
  ]
}
```

## 3.15. グループ情報取得(/batch/groups)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○
1	group_id_list	情報を取得する対象のグループをIDの配列で指定する。	array of int	*1
1	group_unique_alias_list	情報を取得する対象のグループを識別名の配列で指定する。	array of string	*1

\*1: どちらか必須

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	処理結果コード一覧を参照。	string	○
1	groups	配下のパラメータについては、管理者マニュアルのグループインポートのパラメータ一覧を参照。	array of object	*1

\*1: 処理結果コードが正常終了の場合に設定される。

- サンプル

- リクエスト

識別名が sample\_group1 のグループの情報を取得する。

```
{
  "authenticity_token": "*****" (認証トークン) ,
  "group_unique_alias_list": ["sample_group1"]
}
```

- レスポンス  
取得成功

```
{
  "result_code": "0000",
  "groups": [
    {
      "unique_alias": "sample_group1",
      "external_owner": null,
      "child_groups": [
        "sample_group1_1",
        "sample_group1_2",
      ],
      "child_groups": [
        "sample_user1"
      ],
      "name": {
        "en-US": "sample_group1"
      },
      "is_root": true
    }
  ]
}
```

※ ※

## 3.16. ログアウト(/logout)

- リクエスト

階層	ノード名	説明	型	必須
1	authenticity_token	ログインI/Fで返却された認証トークンを指定する。	string	○

- レスポンス

階層	ノード名	説明	型	必須
1	result_code	「正常終了」を表す処理結果コードを返却する。	string	○

## 4. サンプルプログラム

ASFALIS TransServer は、Web APIを利用したサンプルプログラムを同梱しています。このサンプルプログラムは、実際にプログラムを実装する際の参考にしていただくことができます。

サンプルプログラムは下記から入手可能です。

<ASFALIS TransServer導入フォルダ>\k2dashboard\public\samples\web\_api

### 4.1. ライセンスと免責条項

本章で参照するサンプルプログラムは、ASFALIS TransServer Web APIの利用方法の例を示す目的で提供されています。この目的を超える使用について、エリジオンは一切のサポートをいたしかねます。また、本章で参照するサンプルプログラムの使用によって発生した損害について、エリジオンは一切責任を負わないものとします。

### 4.2. EntryAndDownload

EntryAndDownload は、ASFALIS TransServerにジョブを投入し、ジョブが正常終了になるまで待ち、結果ファイルをダウンロードするサンプルです。このサンプルでは、Web APIを以下のように呼び出します。

1. ログイン (/login/authenticate)
2. ファイルセット登録 (/batch/file\_set\_register)
3. ジョブ登録 (/batch/job\_entry)
4. ジョブステータス取得 (/batch/job\_state) : ジョブが完了するまで繰り返し呼び出す
5. ジョブダウンロード (/batch/job\_download)
6. ログアウト (/logout)

下記のプログラミング言語のソースコードが参照可能です。

- C#
  - <ASFALIS TransServer 導入フォルダ>\k2dashboard\public\samples\web\_api\C#\EntryAndDownload
- JavaScript (Node.js)
  - <ASFALIS TransServer 導入フォルダ>\k2dashboard\public\samples\web\_api\JavaScript\EntryAndDownload



本コンテンツに関わる著作権は株式会社エリジオンもしくは原権利者に帰属しています。  
著作権者の承諾なしに無断で改変、複製、転載、再配布、転送、公衆送信、販売、貸与などの  
行為をすることは禁じられています。