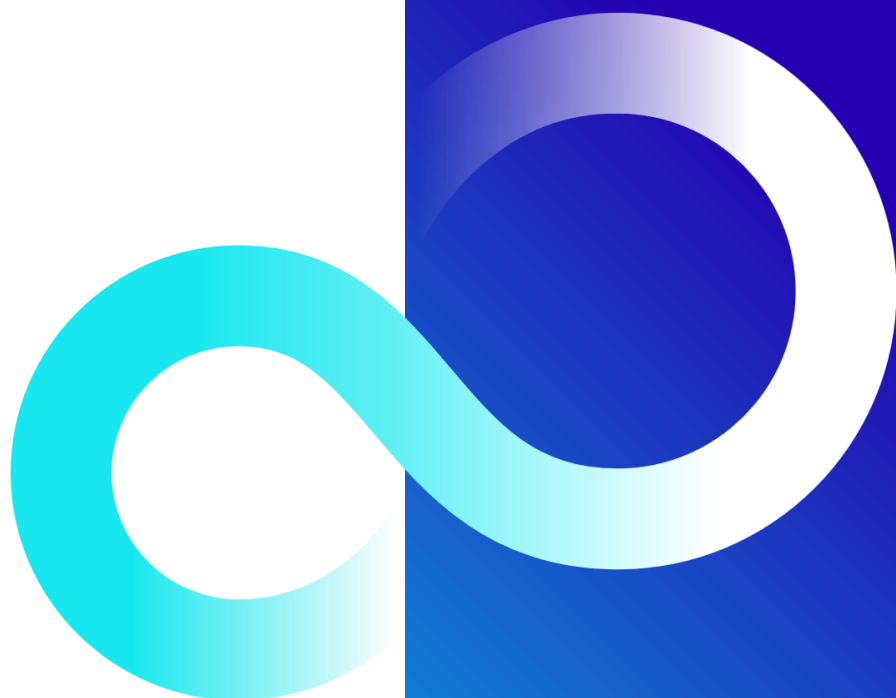


# HPC WebAPI 利用ガイド



2023年10月12日  
富士通株式会社

## 目次

1.	<a href="#">はじめに</a>	3
2.	<a href="#">利用シーン</a>	4
3.	<a href="#">利用開始までのフロー</a>	4
4.	<a href="#">オブジェクトストレージ</a>	4
4.1	<a href="#">オブジェクトストレージ認証情報の取得</a>	4
4.2	<a href="#">利用前の準備</a>	4
4.3	<a href="#">API利用のためのオブジェクトストレージ認証情報設定</a>	5
4.4	<a href="#">アップロード・ダウンロード</a>	5
5.	<a href="#">認証情報とヘッダー情報の用意</a>	8
5.1	<a href="#">API利用のための認証情報とヘッダー情報について</a>	8
5.2	<a href="#">API リファレンス</a>	8
6.	<a href="#">HPC Job操作系API利用</a>	9
6.1	<a href="#">ジョブ実行に利用するファイルの準備</a>	9
6.2	<a href="#">APIでデータをFujitsu クラウドサービス HPCに同期させる</a>	10
6.3	<a href="#">APIでジョブを実行する</a>	12
6.4	<a href="#">APIでジョブの状態を確認する</a>	13
6.5	<a href="#">APIでジョブの実行を中止する</a>	15
6.6	<a href="#">APIでデータをオブジェクトストレージに同期させる</a>	16
6.7	<a href="#">オブジェクトストレージから演算済みデータをダウンロードする</a>	18

# 1. はじめに

本書は Fujitsu クラウドサービス HPCのHPC WebAPIを利用するためのガイドです。  
以下「HPC WebAPI」は「WebAPI」という。

「Fujitsu クラウドサービス HPC」の基本的な利用方法は、[Fujitsu クラウドサービス HPC ユーザーガイド](#) をご参照ください。

## 1.1 関連ドキュメント

- [Fujitsu クラウドサービス HPCユーザーガイド](#)
- [HPC API 認証情報とヘッダー情報](#)
- [APIリファレンス](#)

## 1.2 更新履歴

版数	更新日	更新内容
1.0	2022年11月18日	初版
2.0	2023年6月1日	CaaS HPCとの統合のため、CaaSポータルに関する記述を削除
3.0	2023年8月1日	APIリファレンスとHPC API認証情報とヘッダー情報のURL変更に伴う修正
4.0	2023年10月12日	二フクラウドキュメントのURL変更に伴う修正

## 2. 利用シーン

利用シーンは [Fujitsu クラウドサービス HPC ユーザーガイド の 2.1. 利用イメージ](#) をご覧ください。

## 3. 利用開始までのフロー

利用開始までのフローは [Fujitsu クラウドサービス HPC ユーザーガイド の 3.1. 利用開始までのフロー](#) をご覧ください。

## 4. オブジェクトストレージ

WebAPI利用時のディレクトリ及びファイル操作はオブジェクトストレージを利用します。

### 4.1 オブジェクトストレージ認証情報の取得

オブジェクトストレージ認証情報の取得については [Fujitsu クラウドサービス HPC ユーザーガイド の 5.4.1. アクセスキーとシークレットキーの取得](#) をご覧ください。

### 4.2 利用前の準備

WebAPIで利用するバケットを作成します。オブジェクトストレージでバケットを作成するには、[オブジェクトストレージサービス](#) のAPIを利用します。APIを利用したオブジェクトストレージの操作方法については、[オブジェクトストレージサービスAPI](#) をご覧ください。SDKからAPIを利用する場合は、[オブジェクトストレージサービス SDK for Javaについて](#) をご覧ください。

### 4.3 API利用のためのオブジェクトストレージ認証情報設定

APIを利用してオブジェクトストレージとFujitsu クラウドサービス HPCの間でファイルを同期させるために、ログインノードを使ってオブジェクトストレージの認証情報を設定します。ログインノードへの接続方法は、[Fujitsu クラウドサービス HPC ユーザーガイド の 3.2. ログインノードへの接続](#) をご覧ください。

#### 4.3.1 認証情報管理コマンド「aws-vault」のインストール

ログインノードに接続後、認証情報の保管のためにログインノードに **aws-vault** をインストールします。まず、aws-vaultのインストール先ディレクトリを下記コマンドで作成します。

```
mkdir -p ~/bin
```

ディレクトリ作成後、下記コマンドを実行してaws-vaultをダウンロードします。aws-vaultのバージョンは[こちら](#)を参照し、v6.6.xの最新版をダウンロードします。下記コマンドは、「v6.6.0」をダウンロードする場合のコマンド例です。別バージョンのaws-vaultをダウンロードする場合は、コマンド内の「v6.6.0」の部分を置き換えて実行します。

```
curl -L https://github.com/99designs/aws-vault/releases/download/v6.6.0/aws-vault-linux-amd64 -o ~/bin/aws-vault
```

aws-vaultがダウンロードできたら、下記コマンドを実行してaws-vaultに実行権限を付与します。

```
chmod +x ~/bin/aws-vault
```

次に、下記3つのコマンドを実行し、bashの設定ファイルで環境変数を定義します。

```
echo 'export AWS_VAULT_FILE_PASSPHRASE=' >> ~/.bash_profile  
echo 'export AWS_VAULT_BACKEND=file' >> ~/.bash_profile  
echo 'export AWS_VAULT_PASS_PREFIX=aws-vault' >> ~/.bash_profile
```

以上で、aws-vaultのインストールは完了です。続いて、aws-vaultを利用して認証情報の登録を実施します。

### 4.3.2 「aws-vault」を利用した認証情報の登録

aws-vaultを利用した認証情報の登録を実施します。

まず、ログインノードで環境変数をシェルに反映させるために、下記コマンドを実行します。

```
source ~/.bash_profile
```

次に、aws-vaultを利用して認証情報を登録します。下記コマンドを実行すると、「Enter Access Key ID:」と表示されるため、オブジェクトストレージのアクセスキーを入力します。続いて「Enter Secret Access Key:」と表示されるため、オブジェクトストレージのシークレットアクセスキーを入力します。

```
aws-vault add $(whoami)
```

上記コマンドで誤った内容を入力した場合は、上記コマンドを再度実行し、アクセスキーとシークレットアクセスキーを入力してください。コマンドの再実行により、aws-vaultで管理する認証情報が修正されます。

以上で、オブジェクトストレージの認証情報の登録は完了です。続いて、利用するオブジェクトストレージのバケット情報の設定を実施します。

### 4.3.3 オブジェクトストレージのバケット情報の設定

APIで利用するオブジェクトストレージのバケット情報を設定します。まず、ログインノードで下記2つのコマンドを実行し、バケット名設定用のディレクトリとファイルを作成します。

```
mkdir -p ~/.config/hpc-config  
touch ~/.config/hpc-config/bucket
```

ファイルを作成したら、vimなどのエディタを利用してファイルを編集します。

```
vim ~/.config/hpc-config/bucket
```

上記コマンドで開いたファイルに、利用するオブジェクトストレージのバケットの情報を1行で記載します。バケットの情報は、「s3://」(接頭辞)+「バケット名」の形式で記入します。例えばバケット名が「example」である場合、ファイルには下記のような1行を記入します。

```
s3://example
```

記入した内容を保存し、エディタを終了します。以上で、バケット名設定は完了です。

以上の操作が完了したら、APIを利用してオブジェクトストレージとFujitsu クラウドサービス HPCの間でファイルを同期させることが可能になります。APIの利用方法については、[6. HPC Job操作系API利用](#)をご覧ください。

## 4.4 アップロード・ダウンロード

オブジェクトストレージに対するファイルのアップロード・ダウンロード操作をするには、[オブジェクトストレージサービス](#)のAPIを利用します。APIを利用したオブジェクトストレージの操作方法については、[オブジェクトストレージサービスAPI](#)をご覧ください。SDKからAPIを利用する場合は、[オブジェクトストレージサービス SDK for Javaについて](#)をご覧ください。

## 5. 認証情報とヘッダー情報の用意

### 5.1 API利用のための認証情報とヘッダー情報について

シグネチャーを生成し、WebAPIを利用するための認証情報をヘッダー情報を用意します。

詳細は[HPC API 認証情報とヘッダー情報](#)をご覧ください。

### 5.2 APIリファレンス

本章で用意した認証情報とヘッダー情報を使用して、次章からHPCのJob操作系APIを利用します。

APIリファレンスは[APIリファレンス](#)をご覧ください。



## 6. HPC Job操作系API利用

### 6.1 ジョブ実行に利用するファイルの準備

WebAPIからのジョブ実行のために、ジョブスクリプトの作成、および利用するプログラム・データ・スクリプト等の各種ファイルについてオブジェクトストレージへのアップロードが必要になります。WebAPIによるジョブ実行は、ログインノードでの pjsubコマンド実行に相当します。ジョブスクリプトの作成方法の詳細については、[Fujitsu クラウドサービス HPC ユーザーガイドの4.4. ジョブスクリプトの作成](#)をご覧ください。ジョブ実行時のオプションについては、[Fujitsu クラウドサービス HPC ユーザーガイドの4.5. ジョブ投入](#)をご覧ください。

なお、ジョブ実行時のオプションは、WebAPIからは直接指定できないためジョブスクリプト内で記述しておく必要があります。また、WebAPIでは会話型ジョブは利用できません。そのため、`--interact` オプションは使用しないでください。

プログラム・データ・スクリプト等の各種ファイルのアップロード方法は [4.4 アップロード・ダウンロード](#) をご覧ください。

## 6.2 APIでデータをFujitsu クラウドサービス HPCに同期させる

オブジェクトストレージのバケット内にあるオブジェクトをFujitsu クラウドサービス HPCに同期するには、Start to sync files APIを利用します。詳細は [APIリファレンス Start to sync Files](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

### 6.2.1 同期処理の開始

このAPIでは、リクエストボディで3つのパラメータを指定します。

- path: オブジェクトストレージのバケット内にあるオブジェクトを格納するディレクトリパスを指定します。このパラメータで指定したディレクトリをFujitsu クラウドサービス HPC上に作成し、オブジェクトストレージから同期されたファイルを格納します。
- direction: "from\_online"を指定します。
- delete: オブジェクトの削除を同期するかどうかを指定します。trueを指定すると、オブジェクトストレージで削除したオブジェクトがFujitsu クラウドサービス HPCからも削除されます。falseを指定すると、削除は同期されません。このパラメータを省略した場合、APIはfalseを指定したときと同じ動作をします。

Start to sync files APIのリクエスト例とレスポンス例を示します。この例では、オブジェクトストレージ上のバケット内にあるオブジェクトを、「my\_directory」というディレクトリに同期します。例えば、バケット内に「scripts/my\_script.sh」というオブジェクトが存在するとき、APIを実行することによりFujitsu クラウドサービス HPCで「my\_directory/scripts/my\_script.sh」というファイルが同期されます。

#### ■ リクエストパラメータ例

```
{
  "path": "my_directory",
  "direction": "from_online",
  "delete": false
}
```

#### ■ レスポンス例

```
HTTP/1.1 200 OK

{
  "requestid": 1234567890
}
```

200レスポンスが取得できた場合、ファイル同期処理が開始されています。レスポンスで得られた"requestid"は、ファイル同期処理の状態取得に利用します。

## 6.2.2 同期処理の状態確認

次に、ファイル同期処理の状態をGet Synchronization Status APIを利用して取得します。詳細は [APIリファレンス Get Synchronization Status](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください

ファイル同期処理状態を確認するには、パスパラメータにStart to sync files APIのレスポンスから取得したrequestidを指定します。リクエスト成功時のレスポンスボディでは、2種類のパラメータがあります。

- status: 同期処理の状態を示します。
  - waiting: ファイル同期処理中です。この値であった場合は、時間をおいてから同じrequestidを用いて再度APIを実行してファイル同期処理状態を確認してください。
  - complete: ファイル同期処理が完了した状態です。
  - error: 同期処理に失敗したことを示します。“error: ”の後に続けて失敗の原因が表示されます。間違ったrequestidが指定された場合は、“error: no such ID”となります。
- result: statusが“complete”の場合のみ返されるパラメータです。
  - 0: ファイル同期処理が正常に完了しています。
  - 0以外: オブジェクトストレージの認証情報設定に誤りがある可能性があります。 [4.3 API利用のためのオブジェクトストレージ認証情報設定](#) を確認し、認証情報の修正を実施してください。

レスポンス例を下記で示します。

### ■ レスポンス例：同期処理中

```
HTTP/1.1 200 OK

{
  "status": "waiting"
}
```

### ■ レスポンス例：同期処理成功時

```
HTTP/1.1 200 OK

{
  "result": "0",
  "status": "complete"
}
```

### ■ レスポンス例：同期処理失敗時

```
HTTP/1.1 200 OK

{
  "result": "1",
  "status": "complete"
}
```

### ■ レスポンス例：存在しないrequestidを指定した場合

```
HTTP/1.1 200 OK

{
  "status": "error: no such ID"
}
```

### 6.3 APIでジョブを実行する

HPCでジョブを実行するには、Submit a Job APIを利用します。詳細は [APIリファレンス Submit a Job](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

Submit a Job APIでは、リクエストボディで2種類のパラメータを指定します。

- Jobfile: [6.2.1 同期処理の開始](#) のStart to sync files APIによってオブジェクトストレージからFujitsu クラウドサービス HPC に同期されたオブジェクトのうちの、ジョブ実行用スクリプトを指定します。
- qopt: このパラメータは将来の拡張機能に備えて予約されたパラメータのため、指定は不要です。指定してもAPIの動作には影響ありませんが、指定する場合は“-v”のように文字列型の値である必要があります。文字列以外の値を指定した場合、レスポンスのステータスコードは404になります。

Submit a Job APIのリクエスト例とレスポンス例を示します。オブジェクトストレージのバケット内に「scripts/my\_script.sh」というオブジェクトがあり、Start to sync files API で「"path": "my\_directory", "direction": "from\_online"」としてFujitsu クラウドサービス HPCにオブジェクトを同期した状態の場合、「my\_directory/scripts/my\_script.sh」を使ってジョブを実行することが可能になります。このとき、下記の例のようにリクエストパラメータを指定します。

#### ■ リクエストパラメータ例

```
{
  "jobfile": "my_directory/scripts/my_script.sh"
}
```

#### ■ レスポンス例

```
HTTP/1.1 201 Created

{
  "job_id": 123
}
```

Submit a Job APIの処理が成功すると、レスポンスボディで“job\_id”が取得できます。“job\_id”を利用したジョブの状態確認方法については、次節をご覧ください。

## 6.4 APIでジョブの状態を確認する

### 6.4.1 ジョブ一覧の取得

実行中および実行待ちの全てのジョブの一覧を取得するには、Get Job List APIを利用します。詳細は [APIリファレンス Get Job List](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

JSON形式のレスポンスボディでは、ジョブの状態を示すObject型データを要素とした、array型の配列データが取得できます。実行が完了しているジョブは、ジョブ一覧からは除外されます。実行中のジョブが無い場合は、レスポンスボディが空配列となります。Get Job List APIのレスポンス例を示します。なお、本APIは [fugaku-api](#) をもとに実装されており、ジョブの状態は [fugaku-api](#) に準拠した内容で表示されます。

#### ■ レスポンス例：実行中のジョブが存在する場合

```
HTTP/1.1 200 OK
```

```
[
  {
    "Fugaku:core": "-",
    "Fugaku:e_pol": "-",
    "Fugaku:ec": "0",
    "Fugaku:elapse_limit": "0001:00:00",
    "Fugaku:lst": "RNA",
    "Fugaku:md": "NM",
    "Fugaku:pc": "0",
    "Fugaku:rank": "bychip",
    "Fugaku:reason": "-",
    "Fugaku:sn": "0",
    "Fugaku:v_mem": "-",
    "Fugaku:v_pol": "-",
    "Fugaku:vnode": "-",
    "accept": "2022-10-04T14:09:33",
    "elapse_time": "0000:02:08",
    "group": "apiuser1",
    "job_id": "123",
    "job_name": "my_script",
    "node_require": "2",
    "priority": "127",
    "queue": "def_grp",
    "start_date": "2022-10-04T14:09:33",
    "status": "RUN",
    "user": "apiuser1"
  }
]
```

#### ■ レスポンス例：実行中のジョブが存在しない場合

```
HTTP/1.1 200 OK
```

```
[]
```

## 6.4.2 特定ジョブの状態取得

特定の1つのジョブの状態を取得する場合は、Get Job Detail APIを利用します。詳細は [APIリファレンス Get Job Detail](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

パスパラメータにSubmit a Job APIのレスポンスから取得したjob\_idを指定します。指定したjob\_idのジョブが実行中または実行待ち状態の場合、下記の例のようなJSON形式のオブジェクトデータが取得できます。指定したjob\_idのジョブが存在しない、または実行が終了している場合は、レスポンスのステータスコードは404になります。なお、本APIは [fugaku-api](#) をもとに実装されており、ジョブの状態は fugaku-api に準拠した内容で表示されます。

### ■ レスポンス例：指定したjob\_idのジョブが存在する場合

```
HTTP/1.1 200 OK

{
  "Fugaku:core": "-",
  "Fugaku:e_pol": "-",
  "Fugaku:ec": "0",
  "Fugaku:elapse_limit": "0001:00:00",
  "Fugaku:lst": "RNA",
  "Fugaku:md": "NM",
  "Fugaku:pc": "0",
  "Fugaku:rank": "bychip",
  "Fugaku:reason": "-",
  "Fugaku:sn": "0",
  "Fugaku:v_mem": "-",
  "Fugaku:v_pol": "-",
  "Fugaku:vnode": "-",
  "accept": "2022-10-04T14:09:33",
  "elapse_time": "0000:01:56",
  "group": "apiuser1",
  "job_id": "123",
  "job_name": "my_script",
  "node_require": "2",
  "priority": "127",
  "queue": "def_grp",
  "start_date": "2022-10-04T14:09:33",
  "status": "RUN",
  "user": "apiuser1"
}
```

### ■ レスポンス例：指定したjob\_idのジョブが存在しない場合、または実行が終了している場合

```
HTTP/1.1 404 Not Found

{
  "code": "Client.URLNotFound",
  "message": "A specified URL is not found."
}
```

## 6.5 APIでジョブの実行を中止する

実行中または実行待ちのジョブを中止したい場合は、Cancel a Job APIを利用します。詳細は [APIリファレンス Cancel a Job](#) をご覧ください。APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

パスパラメータにSubmit a Job APIのレスポンスから取得したjob\_idを指定します。指定したjob\_idのジョブが実行中または実行待ち状態の場合、ステータスコードが204のレスポンスが得られ、ジョブの実行が中止されます。指定したjob\_idのジョブが存在しない、または実行が終了している場合は、レスポンスのステータスコードは404になります。

- レスポンス例：指定したjob\_idのジョブが存在する場合

```
HTTP/1.1 204 No Content
```

- レスポンス例：指定したjob\_idのジョブが存在しない場合、または実行が終了している場合

```
HTTP/1.1 404 Not Found
```

```
{  
  "code": "Client.URLNotFound",  
  "message": "A specified URL is not found."  
}
```

## 6.6 APIでデータをオブジェクトストレージに同期させる

プログラムやスクリプトが出力したファイルをオブジェクトストレージに同期するには、Start to sync files APIを利用します。詳細は[APIリファレンス Start to sync Files](#)をご覧ください。

APIの利用に必要な認証情報については [HPC API 認証情報とヘッダー情報](#) をご覧ください。

### 6.6.1 同期処理の開始

Start to sync files APIでは、リクエストボディで3つのパラメータを指定します。

- path: ジョブ実行により出力されたファイルが格納されたディレクトリのパスを指定します。このパラメータで指定したディレクトリの下に格納されたファイルが、オブジェクトストレージのバケット内にオブジェクトとして同期されます。
- direction: "to\_online"を指定します。このパラメータを省略した場合、APIは"from\_online"を指定したときと同じ動作をしてしまうため、省略しないでください。
- delete: オブジェクトの削除を同期するかどうかを指定します。trueを指定すると、ジョブ実行によって削除したファイルがあった場合、オブジェクトストレージのバケットからも対応するオブジェクトが削除されます。falseを指定すると、削除は同期されません。このパラメータを省略した場合、APIはfalseを指定したときと同じ動作をします。

Start to sync files APIのリクエスト例とレスポンス例を示します。Fujitsu クラウドサービス HPCでジョブ実行により出力した「my\_directory/result/log.txt」というファイルをオブジェクトストレージに同期したい場合、pathで"my\_directory"と指定することで、オブジェクトストレージのバケット内に「result/log.txt」というオブジェクトが同期できます。

#### ■ リクエストパラメータ例

```
{
  "path": "my_directory",
  "direction": "to_online",
  "delete": false
}
```

#### ■ レスポンス例

```
HTTP/1.1 200 OK

{
  "requestid": 1234567890
}
```

200レスポンスが取得できた場合、ファイル同期処理が開始されています。レスポンスで得られた"requestid"は、ファイル同期処理の状態取得に利用します。



## 6.6.2 同期処理の状態確認

次に、ファイル同期処理の状態をGet Synchronization Status APIを利用して取得します。APIを利用した同期処理状態の確認方法は、[6.2.2 同期処理の状態確認](#)をご覧ください。

## 6.7 オブジェクトストレージから演算済みデータをダウンロードする

計算結果は、オブジェクトストレージから、オブジェクトストレージのAPIを使って取得します。  
ファイルのダウンロード方法は [4.4 アップロード・ダウンロード](#) をご覧ください。

